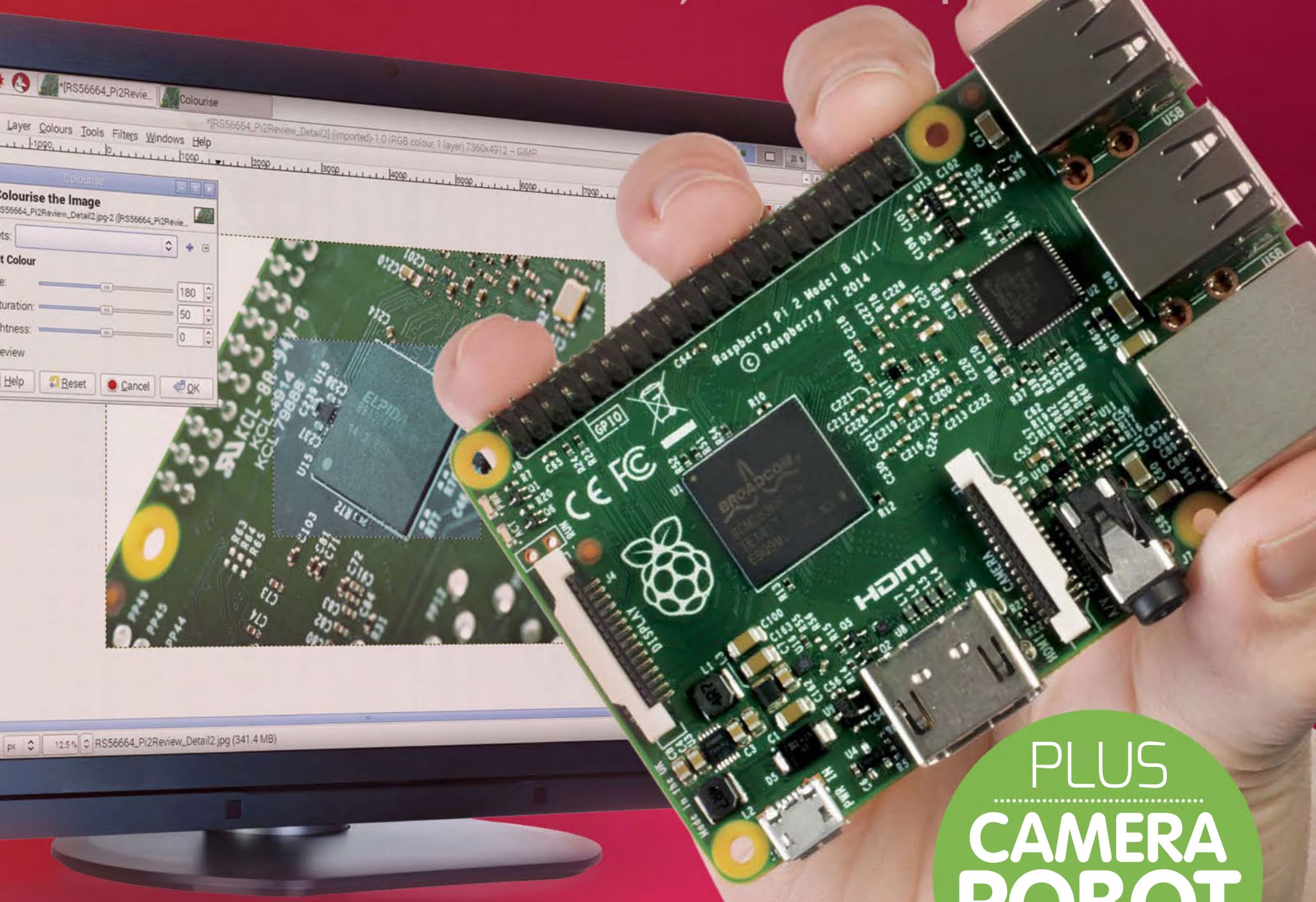
RasPi

DESIGN
BUILD
CODE

16

Get hands-on with your Raspberry Pi



PLUS
CAMERA
ROBOT
TESTED

Make a Pi 2 DESKTOP PC

Welcome

Do you own a Raspberry Pi 2? It's the latest flagship Pi from the Raspberry Pi Foundation and is six times faster than the older models – making it perfect for projects that were previously out of reach. This issue we tell you everything you need to know about the board itself and help you switch over from an old Pi, and then we show you how to turn it into either a home theatre PC or use it as a full-blown desktop PC – it really is that powerful! And do you remember the robot that we built together way back in issues 2 and 3 of **RasPi**? Well, check out the kit from Dawn Robotics that's reviewed towards the end of this issue – it's a pretty ideal present for someone this Christmas!

Gavin Thomas

Editor

Get inspired

Discover the RasPi community's best projects

Expert advice

Got a question? Get in touch and we'll give you a hand

Easy-to-follow guides

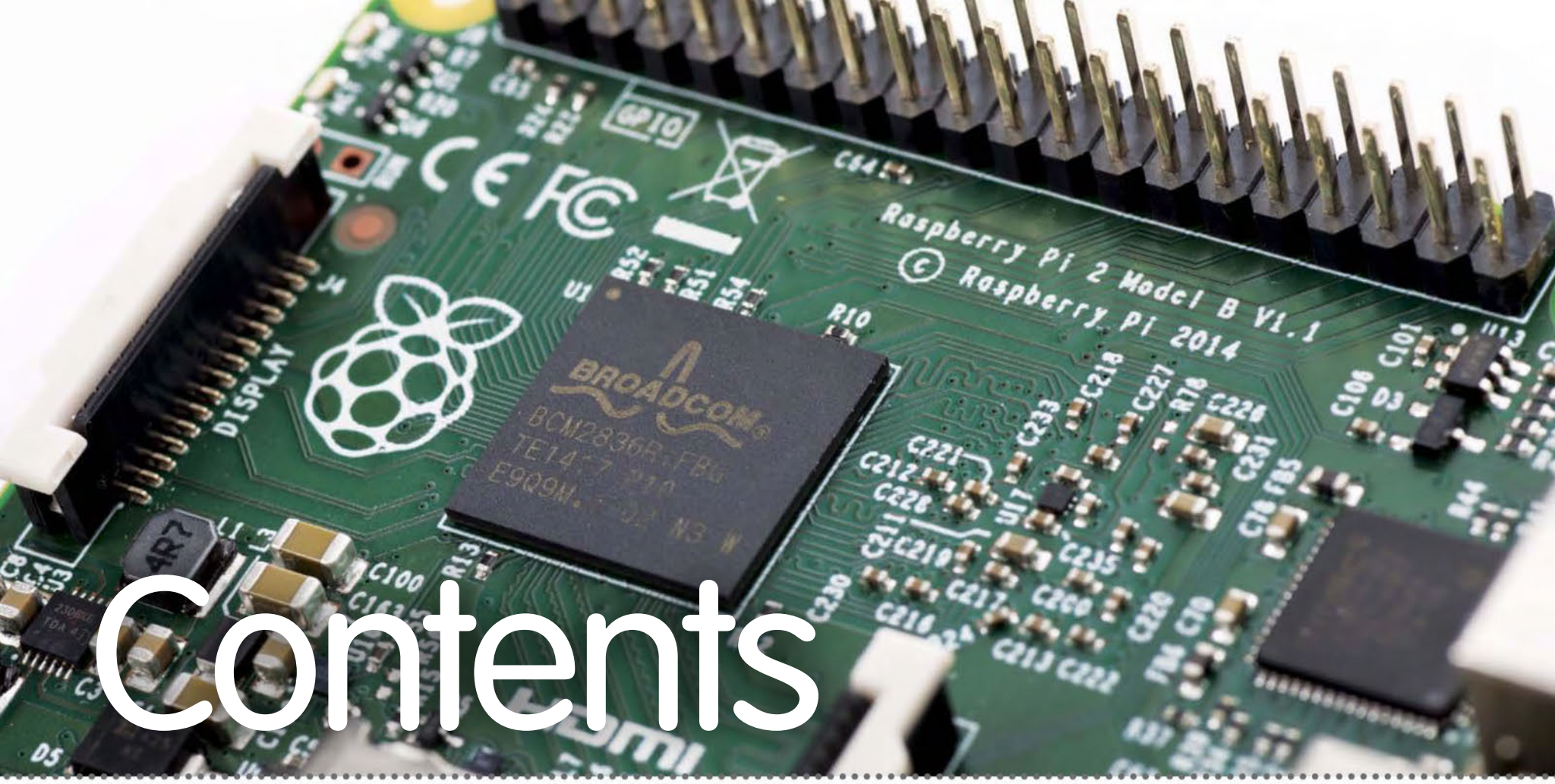
Learn to make and code gadgets with Raspberry Pi

From the makers of

LinuxUser & Developer

Join the conversation at...

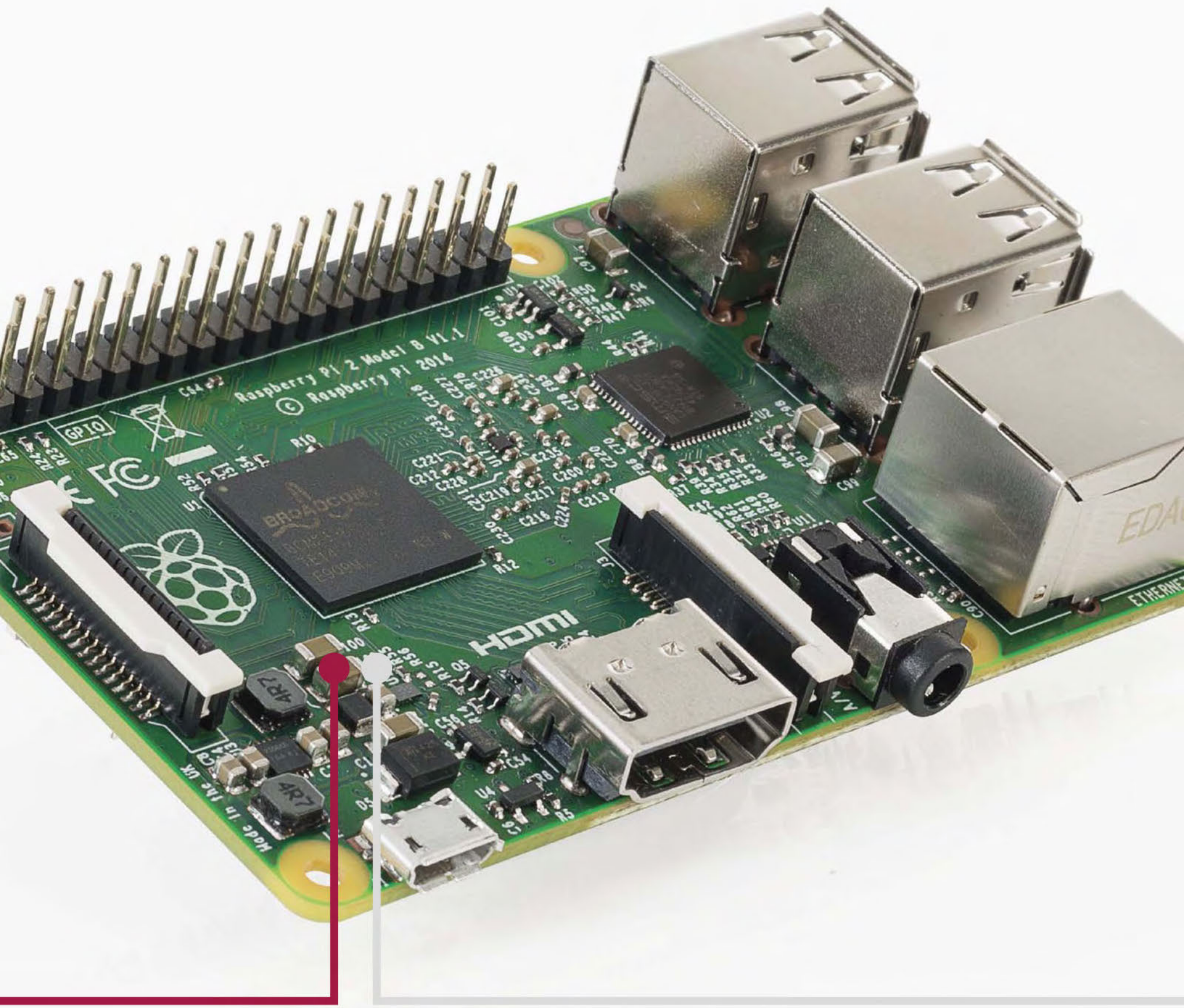
@linuxusermag | Linux User & Developer | RasPi@imagine-publishing.co.uk



Contents

What is the Raspberry Pi 2?

Still confused over what the new version of the Raspberry Pi means to you? We'll fill you in



Q There is another Raspberry Pi then? Is it another update, a brand new thing or something entirely different? I'm getting a bit confused already!

A This time it's the Raspberry Pi 2 Model B – basically a brand new Raspberry Pi.

Q Basically a brand new Raspberry Pi?

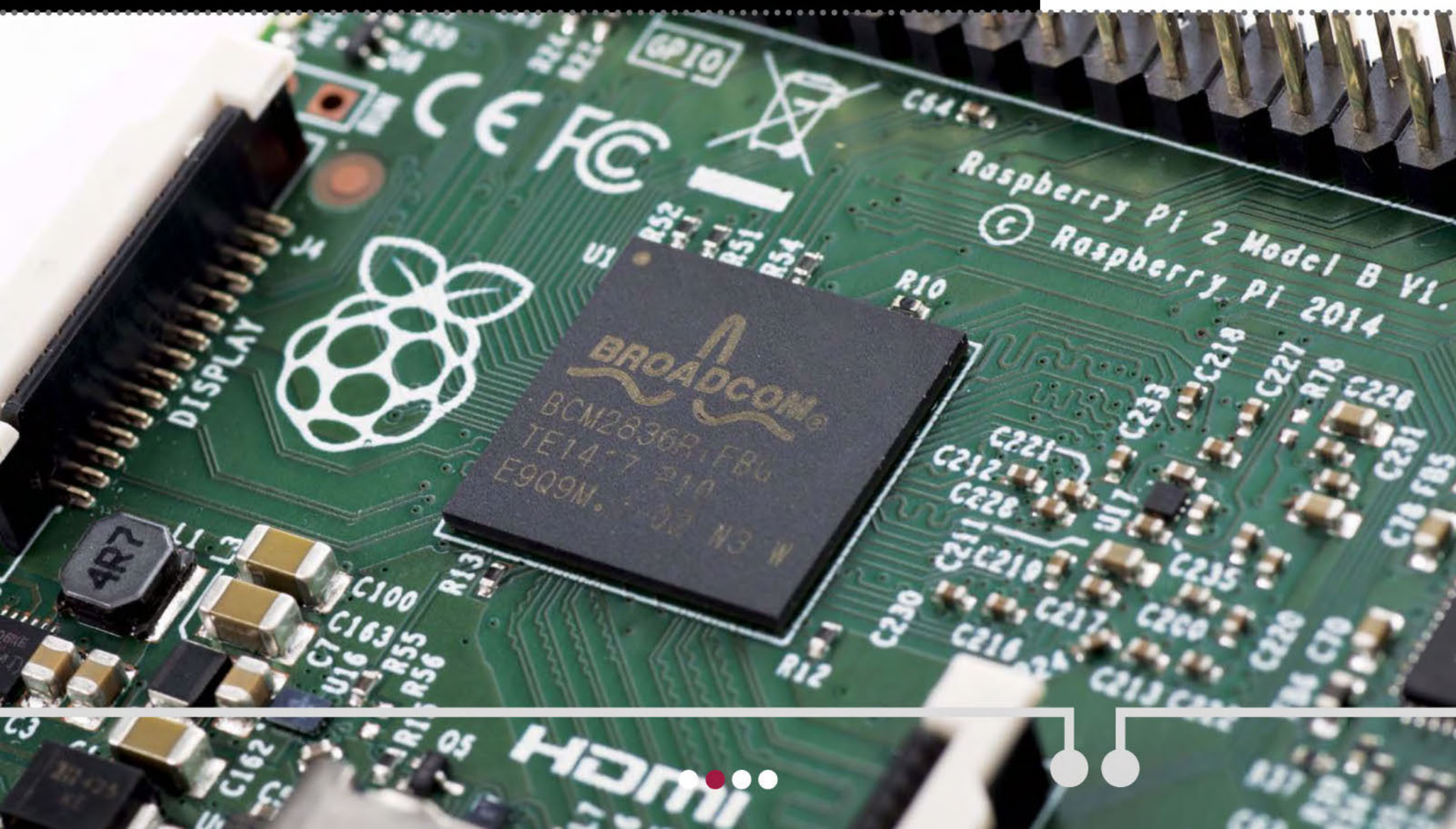
A Well the thing that truly counts, the thing that makes it a Raspberry Pi, is the chip that powers it. The Broadcom BCM2835 chip is literally the core of the original Raspberry Pi, the system-on-a-chip that had the ARMv6 700 MHz processor and the VideoCore graphics chip. This was what was on the compute module from a while back, along with the 512 MB of RAM present in the Pi Model B v2 and B+. Anyway, this chip has been updated and the RAM increased, which is what really makes it the Pi 2.

Q What is the new chip about, then?

A The BCM2836 is a slightly modified version of the 2835 – the main difference is that it now has an ARMv7 quad-

"The BCM2836 is a slightly modified version of the 2835 – the main difference is that it now has an ARMv7 quad-core, 900-MHz chip and 512 Kb of cache"

Below This little processor here is what's boosting up the Raspberry Pi 2



core, 900-MHz chip and 512 Kb of cache. This is paired with an increased RAM of 1 GB and results in a huge power boost to the Raspberry Pi.

Q How much more powerful?

A The Raspberry Pi Foundation estimates that it's about six times more powerful than the original Pi. From the tests that we did, it means that we don't experience slow-down on conventional tasks like updating software or browsing the web for tutorial info, for example.

Q I've just seen a picture of it and it looks no different from the Model B+?

A Yep, the main and only real noticeable difference is that the RAM is now on the underside of the board instead of the top. The B+ redesign itself was actually made to accommodate the Pi 2 hardware, releasing a sort-of stop-gap device that also meant that manufacturers were able to get cases out into the wild before the 2 was a actually released.

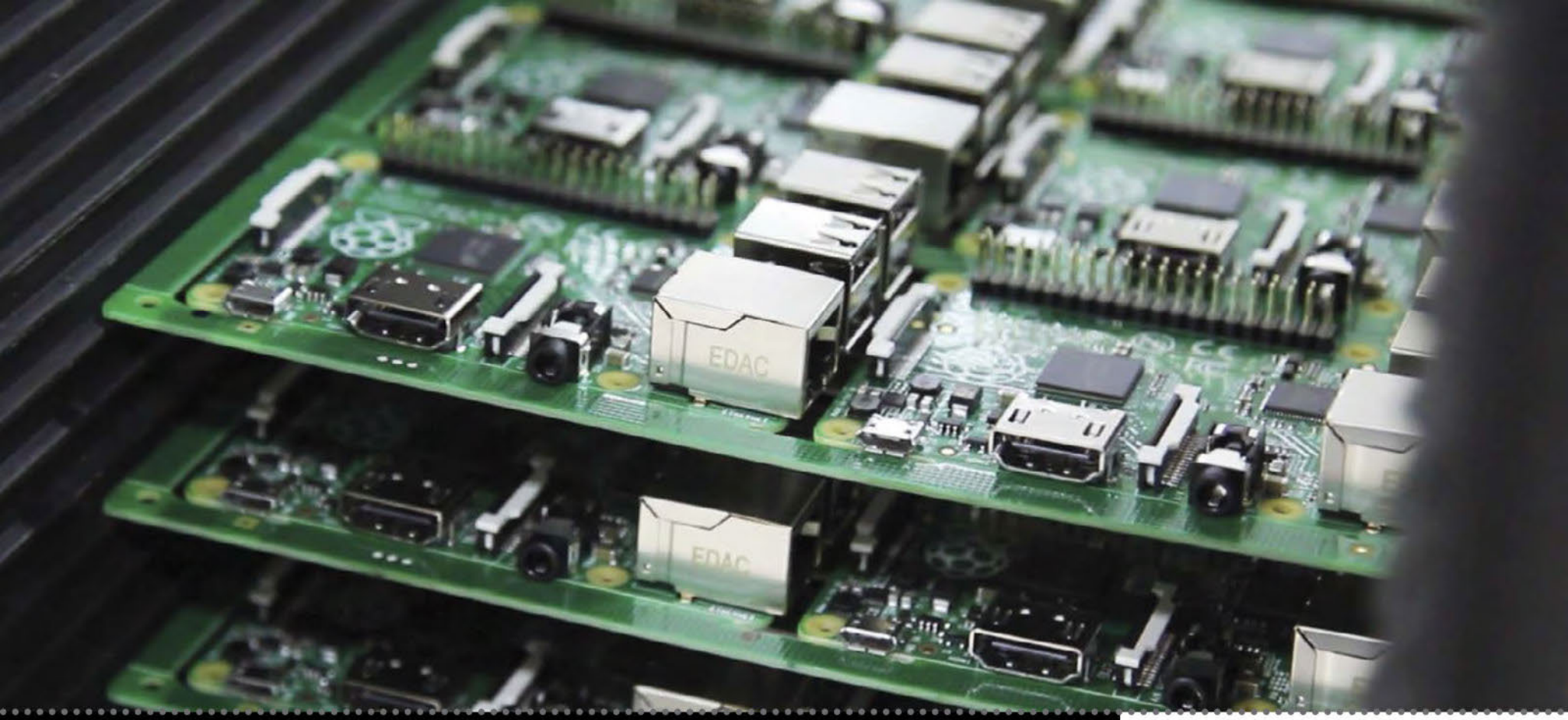
Q Does it have Wi-Fi?

A No, there is no Wi-Fi built in. Apparently that would have been difficult to accomplish. Instead, they've got a Pi-branded Wi-Fi dongle that is very cheap and very powerful.

Q Does everything work on it that worked on the original Raspberry Pi?

A Basically, yes. It's best to grab the updated Raspbian that works better on ARMv7, but as the software itself uses the same packages and the same ports, there should be no problem with the way it works. The only issues you might come across are code snippets that take into account a slower Raspberry Pi, but those can be fixed.

"The Raspberry Pi Foundation estimates that it's about six times more powerful than the original Pi"



Above It wasn't always the case, but the Raspberry Pi is now made in Wales here in the UK

Q Will the old Raspberry Pi models not be supported any more?

A They absolutely will be – the Raspbians and such will still work on the old Pis for the time being, with no plans to end support. New software may require the Pi 2, but the Foundation can't really control that.

Q What about the Model A – will there be a new one of those as well?

A At the moment, the Pi Foundation say there will not be a Model A. But they also said a Raspberry Pi 2 probably wouldn't be until 2017 when they were working on it for the February release this year, so we'll see.

Q Give me the damage, then – how much is the new Raspberry Pi 2 going to cost me?

A It's staying at the same price – $35 or about £25 – and for a much more powerful piece of kit as well. It means that the upgrade is cheap and affordable. You can grab it from the same places as well, like the element14 and RS Components stores.

Migrate to the Pi 2

Move your files and settings over to the Raspberry Pi 2 and make the upgrade as painless as possible

The Raspberry Pi 2 is the new flagship model and should definitely be on your Christmas list if you don't have one. The ultimate upgrade to the Raspberry Pi brings more power and better options for your existing projects, while also opening the way to new and exciting projects. Due to a different processor architecture, though, you can't just put your current Raspbian SD card into the new Pi and expect it to work.

It's not very hard to transfer all the important files and settings over to the Raspberry Pi 2, though, and we'll show you how to make the upgrade in this tutorial.

THE PROJECT ESSENTIALS

SD card reader

MicroSD to SD card adapter

Raspbian 2 image
raspberrypi.org/downloads

01 Clean up Raspbian

Now is the perfect time to do some probably overdue spring cleaning. Delete any unneeded files and pictures, perform a `sudo apt-get dist-upgrade` to make sure the packages are the same as the new Raspbian, then uninstall whatever software you don't need.

02 Copy package list

You can make a list of the packages you have installed and then use this list to install the same packages onto your new Pi. Do this by going into the terminal and typing:

```
dpkg --get-selections > installed-software.log
```

03 Prepare the new Raspbian

On a second, spare microSD card, download the image for the recently updated version of Raspbian from the official site (**https://www.raspberrypi.org/downloads**) and write it to the SD card using dd:

```
$ dd bs=1M if=raspbian2.img of=/dev/[SD card location]
```

"You can't just put your current Raspbian SD card into the new Pi and expect it to work"

04 Transfer home folder contents

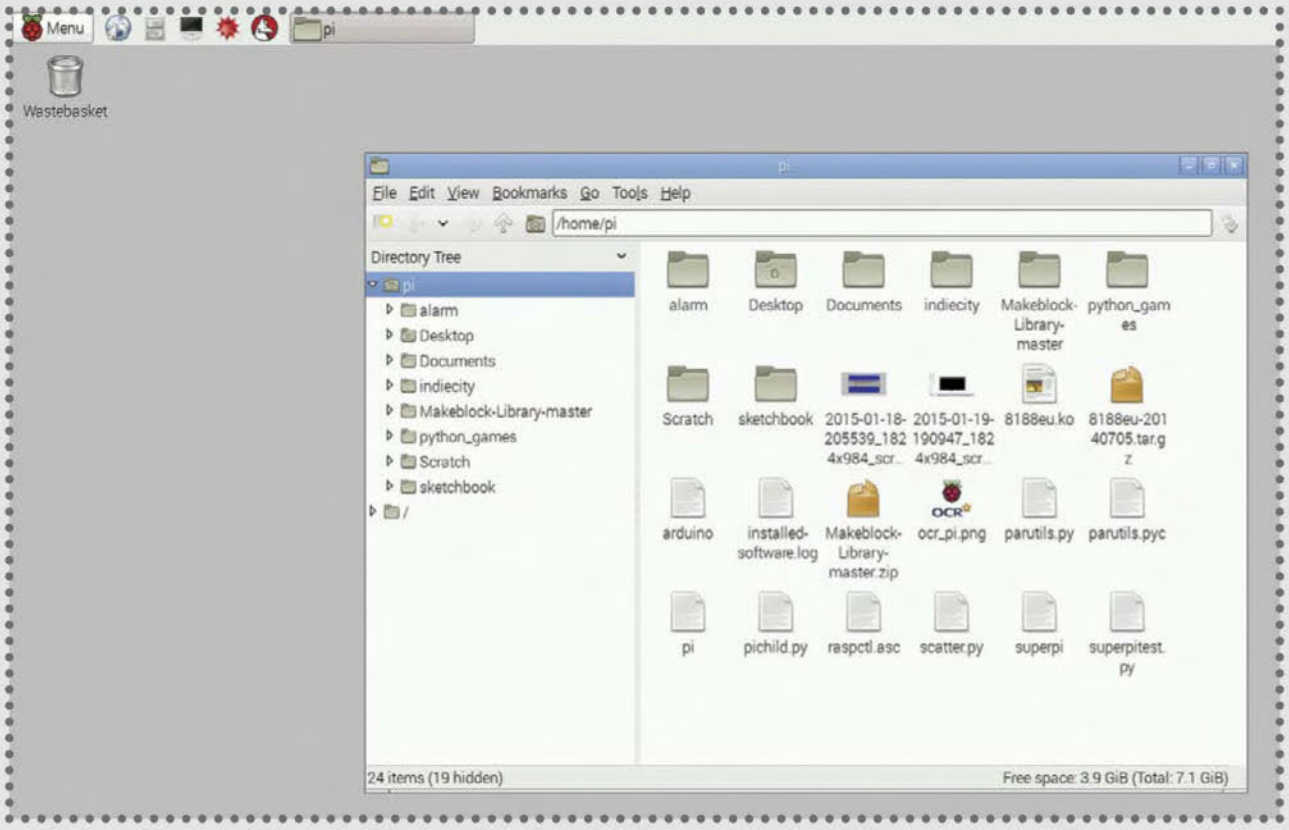
Grab the contents of the home folder from your old Raspbian and either store them on your PC or put them straight onto your new Raspbian SD card if you can have two cards in one PC. Create a backup of this existing card and then re-use it however you want.

05 Set up the new Raspbian

You'll get to raspi-config when you start up the new Raspbian. Perform the usual updates that you'd want, such as turning the desktop on at boot, enabling the camera and expanding the filesystem. Finish and then reboot the Pi.

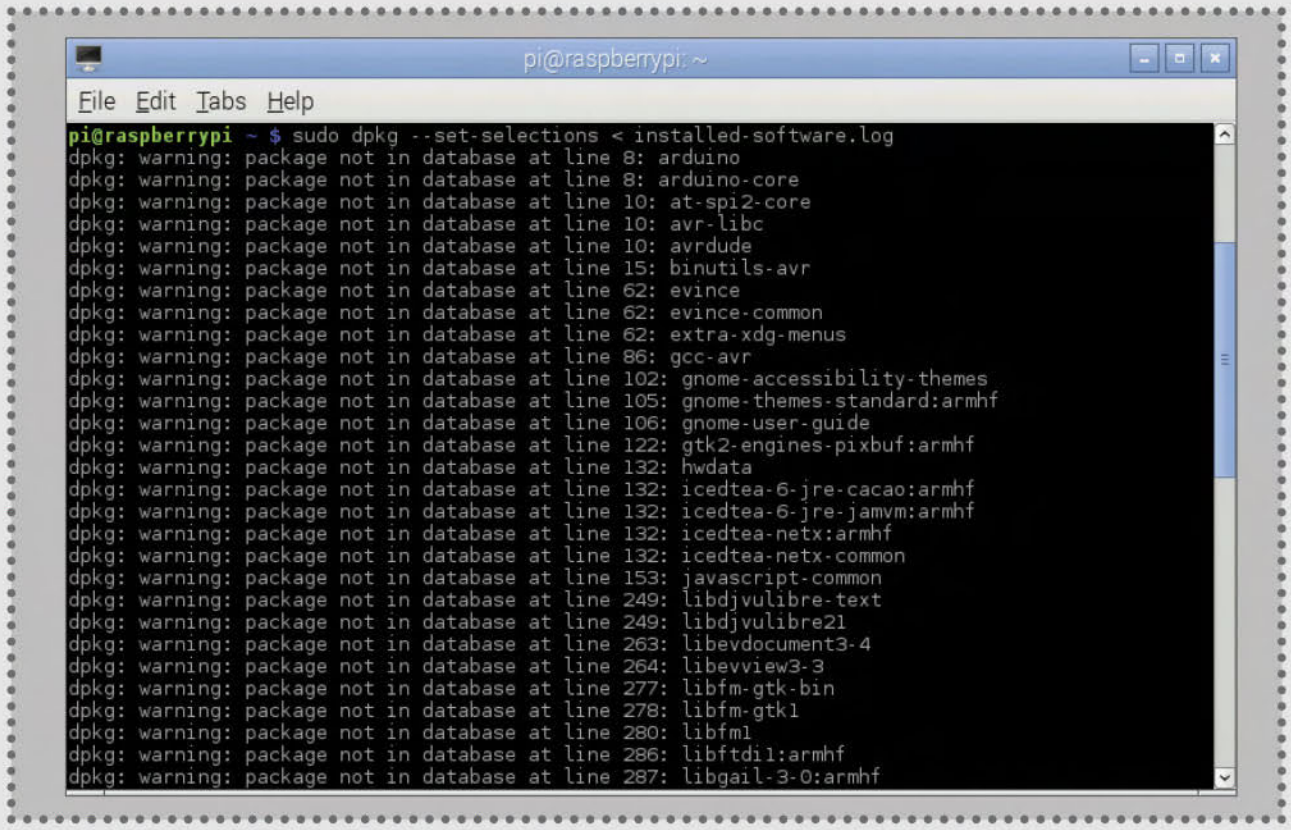
06 Check the files

Make sure all the files have properly transferred over, including the package list. If it was just the home folder files then it should be easy to double-check them. Move anything you've put in there to the folders that they now need to go in.



Left This is an excellent opportunity to clear out those old zips and scripts

"You'll get to raspi-config when you start up the new Raspbian. Perform the usual updates that you'd want, such as turning the desktop on at boot"



Left If you see a lot of errors like these, go through and manually edit the list then retry

07 Restore the files

Open the terminal to restore the packages. Some may not work, so you might have to start manually editing the list of everything to install to remove them – most packages will already be installed as well. Restore with:

```
$ sudo dpkg --set-selections < installed-software. log && sudo apt-get dselect-upgrade
```

08 Upgrade files

Some files may require an upgrade to a newer version after you perform the transfer. To do this, do the normal `sudo apt-get update` followed by a `sudo apt-get upgrade` to make sure the files are back up to date.

09 Enjoy your new Raspberry Pi!

Now you should have successfully upgraded and you can get back to doing existing projects or starting new ones. Next, we'll take a look at how to set up your Raspberry Pi 2 as a HTPC, and even how to use it as a replacement for a traditional desktop PC.

"Next, we'll take a look at how to set up your Raspberry Pi 2 as a HTPC, and even how to use it as a replacement for a traditional desktop PC"

Make a Pi 2 HTPC

Finally create a more powerful and capable HTPC using the Raspberry Pi 2 and the excellent OpenELEC project

We know people who just have a Raspberry Pi for XBMC, now called Kodi. It's a great idea and a great use for the Pi – it works just well enough that you can easily play media locally or over the network. The biggest issue came with GUI response on the original Model Bs, and a lack of USB ports for connecting up everything that you want.

While optimisation over the last few years has helped, the leap to Raspberry Pi 2 has basically solved all of these problems by giving you much more powerful hardware to play with. So if you're looking to upgrade or finally take the plunge, this handy guide will help you create the perfect Raspberry Pi 2 home theatre PC.

THE PROJECT ESSENTIALS

OpenELEC
http://openelec.tv
HDMI cable
USB IR receiver
IR remote
Case
Power supply
USB storage

01 Choose the software

In the past, Pi HTPCs were just a choice between RaspBMC and OpenELEC. However, RaspBMC is on a bit of a hiatus and OpenELEC is your best bet for getting the most up-to-date software. There's not a massive difference between the two, as they both run XBMC.

02 Get the software

Head over to **http://openelec.tv** and look for the Download section. There's a specific Raspberry Pi section which is split up into the original (ARMv6) Pi and the newer Raspberry Pi 2 (ARMv7). Grab the image file from this page for the Pi 2.

03 Install to card

Open up the terminal and use `fdisk -l` to determine where your SD card is located on your system. Something like /dev/sdb or /dev/mmcblk0 will be

> "In the past, Pi HTPCs were just a choice between RaspBMC and OpenELEC. However, RaspBMC is on a bit of a hiatus"

Left The OpenELEC interface is perfectly suited to your remote control app or device

ideal. Navigate to the image using `cd` and install it with dd using:

```
$ dd bs=1M if=OpenELEC-RPi2.arm-5.0.5.img of=/dev/mmcblk0
```

04 First boot

Plug in your Raspberry Pi, either to your TV or to another screen just to begin with, and turn it on. OpenELEC will resize the SD card partitions and write a few extra programs before finally booting into Kodi.

05 Configure Kodi

Go through the basic wizard to get through the interface – if you are connecting via wireless you will need to go to OpenELEC in the System menu and activate the wireless receiver before selecting your network and then entering your password..

06 Add network shares

You can stick a portable hard drive or USB stick into the Pi for storage, but the best method is really to stream over the network. Go to 'File manager' under System and 'Add source'. Go to Browse and choose

"You can stick a portable hard drive or USB stick into the Pi for storage, but the best method is really to stream over the network"

Left Keep all your videos saved to a hard drive that's accessible on your home network

your network protocol to browse the network, or alternatively, add it manually.

07 Build your media centre

Placement of your Raspberry Pi is important. As it's going to be out all the time, we highly recommend getting a case for it – the Pibow cases from Pimoroni are quite well suited for this type of use as they are sturdy and can be attached to the rear of some TVs.

08 IR sensors and controllers

Kodi can be controlled with a number of different things – including USB game controllers and compatible IR sensors. We've used FLIRC in the past, but if you have your Pi behind the TV, you'll need a sensor on a wire that can stretch to a useful position.

09 Future updates

OpenELEC has the excellent ability to update itself without needing you to reinstall it every few months, meaning you won't need to do much maintenance on it at all. Now you can sit back and enjoy your media much easier than before.

"OpenELEC has the excellent ability to update itself without needing you to reinstall it every few months"

Make a Pi 2 desktop PC

Use your Raspberry Pi as a desktop replacement PC thanks to the increased power of the Raspberry Pi 2

You've heard all about the Raspberry Pi 2's increased power over its predecessor by this point. More CPU cores and more RAM making it six times faster is an impressive number, and you can see the actual changes that it makes to the experience by just diving in and running a few programs.

This power actually enables you to conduct a very simple project that was just out of reach for the original Raspberry Pi: a Raspberry Pi desktop PC.

All the components for it were available, but the Pi was just a little too slow to properly give a fluid desktop experience. Now with the improved resources, many of the restrictions are gone – enough of them to be able to build a Pi desktop. So grab a Pi 2, and perhaps your official Raspberry Pi case, and we'll get started.

THE PROJECT ESSENTIALS

Raspbian
raspberrypi.org/downloads
Keyboard
Mouse
Wireless dongle
Case
Monitor
Powered USB hub

01 Get Raspbian

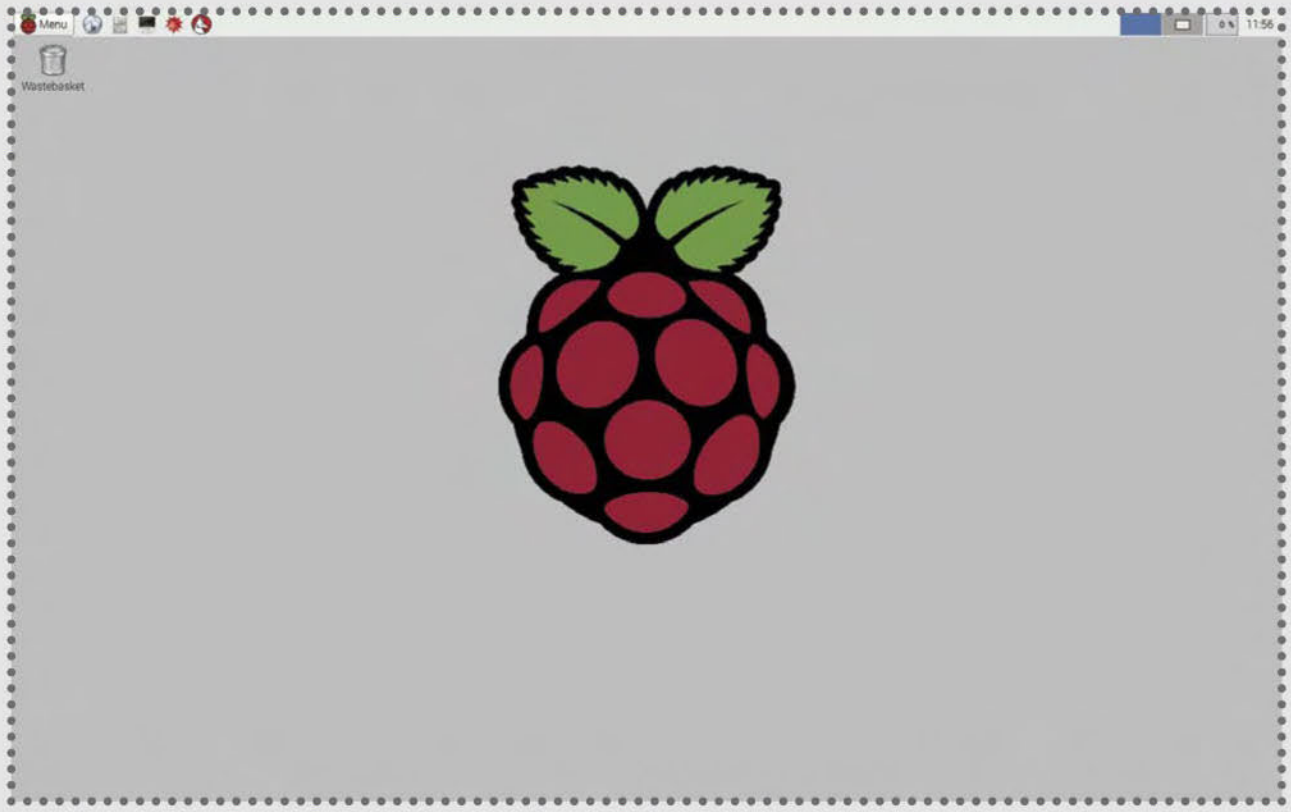
We will be using Raspbian for our desktop Pi. Not only is it simple to obtain and easy to use, but it is supported by the Raspberry Pi Foundation and community, which means it's going to be the most flexible operating system with the most choices for a desktop. Download it from: **www.raspberrypi.org/downloads**.

02 Install Raspbian

Once Raspbian is downloaded, you can install it to your SD card. Put the micro SD into an SD card reader and connect it to your main system (a PC or laptop). Open up the terminal, `cd` to the location of the image and then use:

```
$ sudo dd bs=1M if=raspbian.img of=/dev/
[location of SD card]
```

"This power actually enables you to conduct a very simple project that was just out of reach for the original Raspberry Pi"



Left Booting to the GUI is a must for your Raspberry Pi 2 desktop PC

03 Setup options

On first boot there will be some setup stuff that it is necessary for you to go through. The most important things to do for this desktop are to first hit 'Enable Boot to Desktop' and then to extend the installation to fill the entire SD card. After you have done that, do anything else that you want to do in these menus and then reboot before moving on to the next step.

04 First boot

You will boot into a fresh version of Raspbian with the newer interface and default apps available to use. From here you can start using it as normal if you wish, but it is worth noting that there are a few extra things that you should do to make it truly desktop worthy.

05 Software updates

Our first step is to perform an upgrade on the system to make sure it's all up to date and working properly. To do this, open up the terminal from the menus and use:

```
$ sudo apt-get update
```

"First hit 'Enable Boot to Desktop' and then extend the installation to fill the entire SD card"

… to refresh the software list, followed by the next command to then upgrade to newer software:

```
$ sudo apt-get upgrade
```

06 Firmware upgrade

While we're updating, it's a good idea to upgrade the firmware on the device. Still in the terminal, you'll want to activate the firmware upgrade software with:

```
$ sudo rpi-update
```

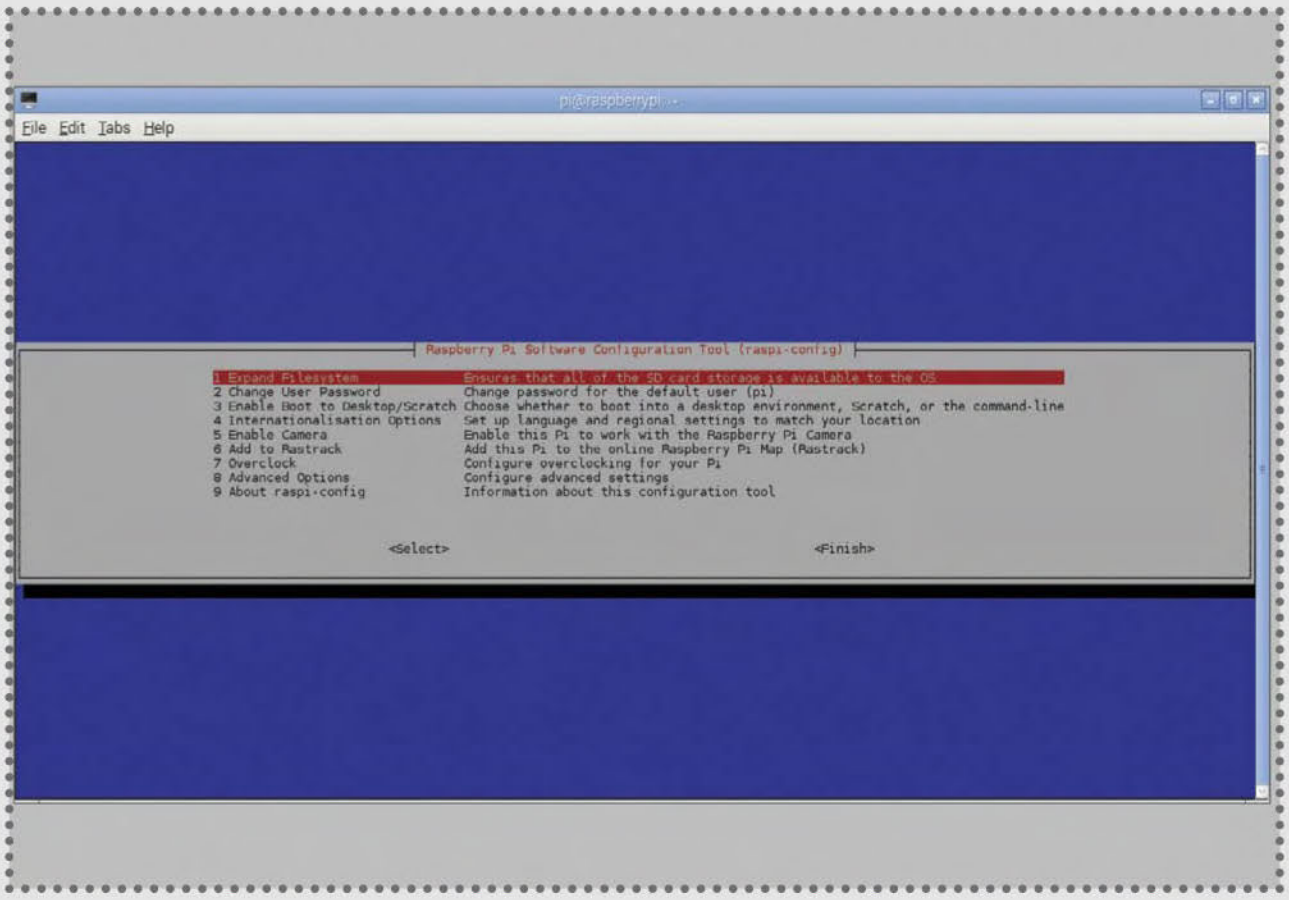
07 Extra configuration

At this point, you might want to tweak the Pi a little further. To bring up the initial configuration screen, you'll need to go back into the terminal and launch it with:

```
$ sudo raspi-config
```

08 Advanced options

From here you can activate some extra options that you might need in the future. Enabling the Pi camera driver is a really good first step, and you can even



"Activate some extra options that you might need in the future. Enabling the Pi camera driver is a really good first step"

Left If you're experienced, and/or brave, play around with the overclocking to boost performance

have it boot to Scratch if you want to focus on fun game development. Otherwise, there are also some overclocking options that you can consider if the system starts getting slow for you.

09 Accessorise your Pi

Just setting up the operating system on the Raspberry Pi is only a small part of the process – we also have to consider the hardware surrounding it that will actually make it usable as a desktop replacement.

Left You don't need anything fancy – a spare wired keyboard will be ideal for this

10 Human input devices

Standard USB keyboards and mice are best suited for this task, much more so than a lot of the wireless keyboard and mouse combos that are popular among Pi users. However, don't try and save on USB ports by getting a keyboard that has USB connections of its own: the Raspberry Pi cannot power USB hubs, even just two on a keyboard.

"Standard USB keyboards and mice are best suited for this task, much more so than a lot of the wireless combos"

Left You'll want to use a monitor that outputs at 1920x1080

11 Monitor to see

The Pi can output a maximum of 1080p, which in normal display terms is 1920 x 1080. While it only outputs in HDMI, a lot of modern monitors do have an HDMI input. If you don't want to get a brand new monitor, though, you can always get a HDMI-to-DVI or HDMI-to-VGA adapter.

12 Case for protection

The Pi is pretty sturdy and we'd be lying if we said we didn't regularly keep ours out of a case. However, it's not indestructible. While we often do a lot of different projects involving accessing different components, a desktop Pi doesn't require this level of access. We like the Pimoroni Pibow cases, but there are several other secure, protective alternatives, like the official one.

"While we often do a lot of different projects involving accessing different components, a desktop Pi doesn't require this level of access"

13 Wireless for Internet

While some people are fine using wired connections, not everyone has that luxury. Wireless dongles are a perfect fit for the Pi, especially now they're almost no larger than the USB port themselves. However, not just any dongle will work and you'll have to check against this list to make sure that you get a compatible one: **http://elinux.org/RPi_USB_Wi-Fi_Adapters**.

> "Not just any Wi-Fi dongle will work and you'll have to check to make sure that you get a compatible one"

14 Anything else?

Our standard desktop PC setup is complete, with one USB port to spare. You can use that single port for USB sticks or portable storage, or you can invest in a powered USB hub in order to give yourself more connectivity options. Otherwise, investing in a good 2A power supply will help ensure you're never short on power for anything.

15 Adding extra software

We're not quite done getting our Raspberry Pi 2 desktop ready just yet. We need to add some extra software to make it feel more like a real desktop. While we already have a browser installed and some of the basics, the first other piece of software we should add is an office suite.

16 Work with LibreOffice

LibreOffice is not installed by default, but as the premier open source office suite, or Linux office suite in general, it is readily available on Raspbian, which is useful. Open up the terminal and install it with the following:

```
$ sudo apt-get install libreoffice
```

17 GIMP for photos

This is the big one – while the original Pi was not quite able to handle LibreOffice, but kind of managed to load certain things, it was useless trying to use GIMP. Now GIMP works just fine, although more complex tasks might make it slow down just a touch. Install it with:

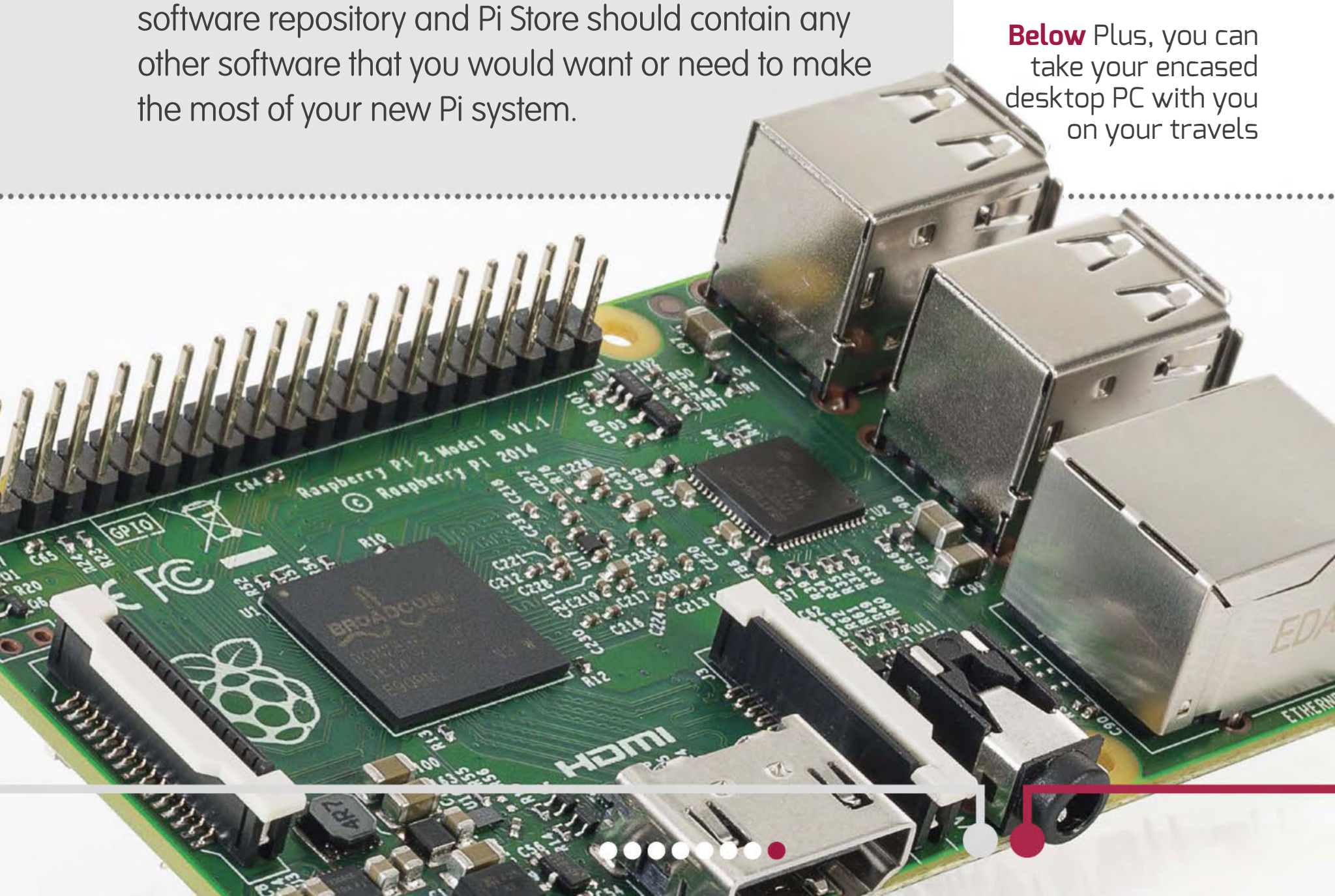
```
$ sudo apt-get install gimp
```

18 XiX for music

If you need to listen to music while you work, one of the best pieces of software to check out is XiX. It's available on the Pi Store. It's a free download and you can find the Pi Store in the Pi menus to install it.

19 Pi for desktop

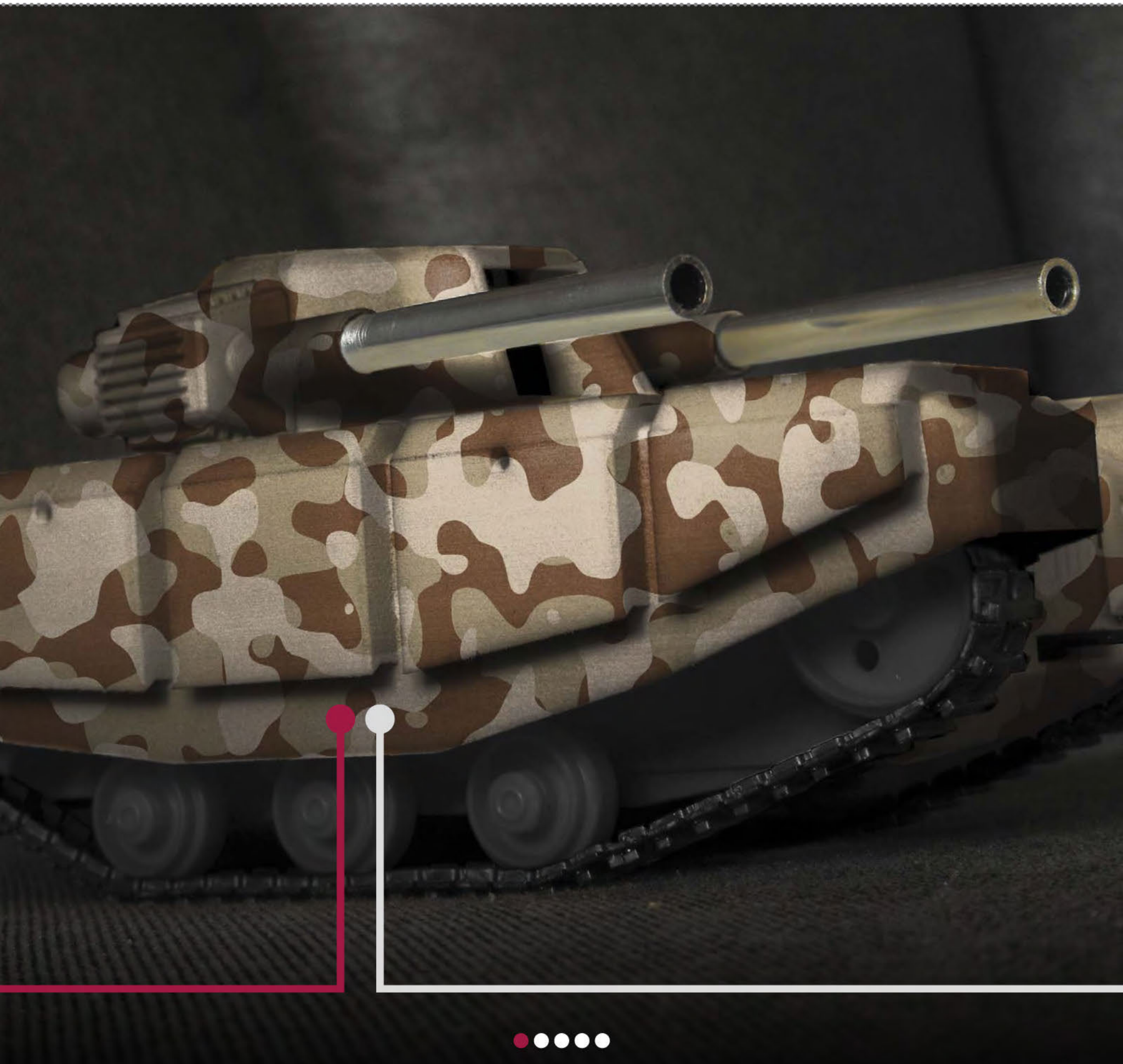
Now we are set up, you can start properly using your Raspberry Pi as a desktop system, while still making use of the educational capabilities when need be. The software repository and Pi Store should contain any other software that you would want or need to make the most of your new Pi system.



Below Plus, you can take your encased desktop PC with you on your travels

PiPanther

Battle your friends in augmented reality using PiPanther, the 3D-printed tank with customisable plugins



Tell me more about your hackable gaming tank – it looks awesome!

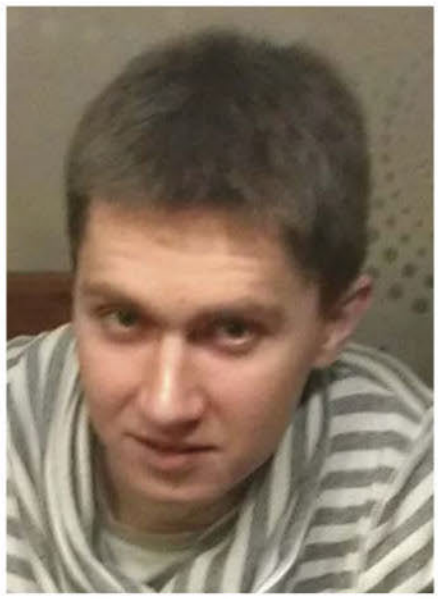
The PiPanther robot is a smart, fast vehicle that you control using the Ubotia app on your smartphone. As it explores rooms, you can set up battlefields anywhere and fight your friends using the Ubotia software. Plus, you can customise the tank with your own attachments. But there's just 15 minutes to go now *[Ed: PiPanther's Kickstarter clock is counting down]* and we only have 22% [of €5,000/$5,900]. It's not enough, unfortunately.

Have you got any further plans for PiPanther, once the Kickstarter campaign has come to an end?

Well it's not 100 percent ready – I mean, it is ready but it needs a little more work. When working on it I found several issues that needed to be fixed, both mechanical and electronic. My colleague and I (the electronics guy) spent hundreds, if not thousands of hours on this project but still something more needs to be done. To be honest, the biggest problem is the casing – this white casing is 3D-printed, but because I designed it using relatively thin and highly precise elements, I couldn't use the cheaper 3D printers that you can buy for a few hundred pounds. Instead I had to order the casing in from a specialised company that used selective laser sintering. So it's more expensive and the casing is very, very precise, but that expense is not acceptable for the production of the PiPanther on a large scale.

So what makes up the PiPanther – what are the main components?

Currently, the chassis consists of four elements. Then there's a Raspberry Pi, our PiPanther mainboard on top of that, two DC motors, two gearboxes, the tracks, one



Michał Okulski is a professional C# software developer with a masters degree in Computer Science from Adam Mickiewicz University. He loves to work on projects combining innovation, robotics, electronics and mechanics.

servo to rotate the turret. Then inside the turret there's a mounted USB hub, because I designed PiPanther to use the Raspberry Pi Model A+, since it's smaller and cheaper, but it only has one USB port. So on the turret you have two USB plugins and then the third USB port is for the Wi-Fi. We also have a Li-Po battery at the bottom of the chassis.

Can you tell me more about the PiPanther mainboard?
We designed it ourselves – it's half the size of the Raspberry Pi and on it you have a DC-DC converter, which supplies 5V to the Pi, powerful motor drivers, a servo controller and a USB charger, so you can charge PiPanther directly using a microUSB cable – just plug it in, leave it for three hours and it's charged.

If you like
Find out how to make your own PiPanther plugin and see a breakdown of the plugin architecture over at **ubotia.com/posts/plugin-specification**. Keep an eye on the main PiPanther page for any updates on the project.

Below Those guns on the side of the turret are replaceable plugins, and you can make your own

This mainboard is going to be available separately for use in custom projects – is that right?

Exactly. To drive a tank, you need to drive two motors – the left track and the right track – but instead of using tracks you can just use two wheels, so you can build your own wheeled robot instead. The steering is exactly the same. If you want to go forward then you power the two motors and if you want to turn then you just regulate power between the left and right motors. So you can use this board to drive your own robot – just plug into your motors and use the same mainboard, but with the Raspberry Pi and Ubotia software. The main part (and the hardest part) of the Ubotia software was to provide live video streaming from the PiPanther to the mobile interface. I made that at a time when nobody else was able to do it, but unfortunately in the meantime, several other robots were developed with video streaming as well. I didn't have too much time because I have a small child, but you have to believe me – I was the first one to create a live video stream from the Raspberry Pi to Android at that time! We achieved a video delay of less than one second, so you can drive it around in real time.

So how much of all this is being handled by the Raspberry Pi?

The Pi runs Arch Linux and on that we have our PiPanther software, which is responsible for talking to the PiPanther mainboard using the serial interface, USART. Then that is used to control the motors, turret, status LEDs and give feedback from the USB charger. That software is written in C++ and also consists of a plugin controller. So if you have plugins then you can put in shared libraries that are loaded at runtime by this software, and using the API you

Further reading

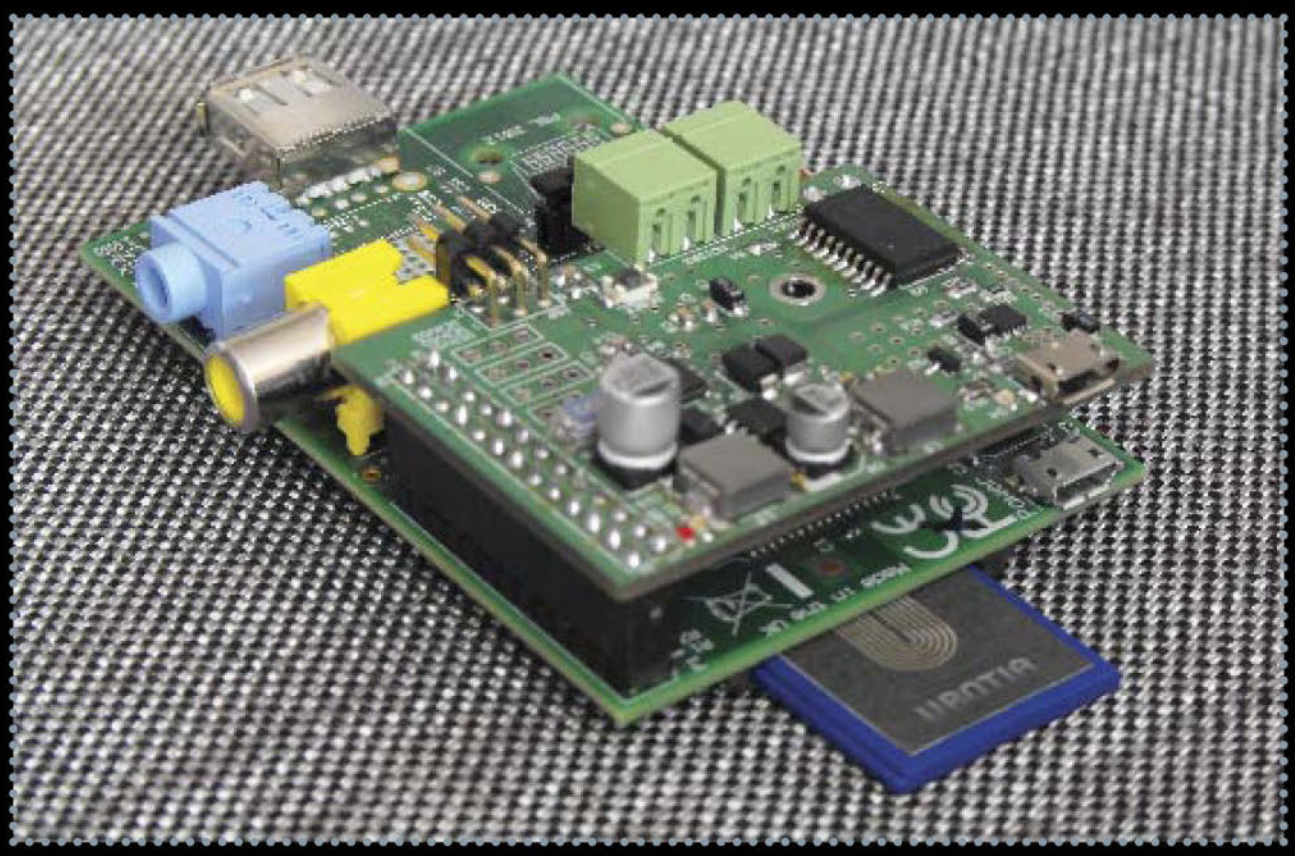
If you're thinking of making your own custom Ubotia robot, you can find our guide to building and controlling Raspberry Pi-powered cars in Raspberry Pi The Complete Manual (**bit.ly/1xg2vuQ**), along with plenty more besides.

can control your own attached plugins using standardised JSON-based protocols.

Your website says that these custom plugins could affect Ubotia gameplay – a robotic arm for a capture the flag match, for example.

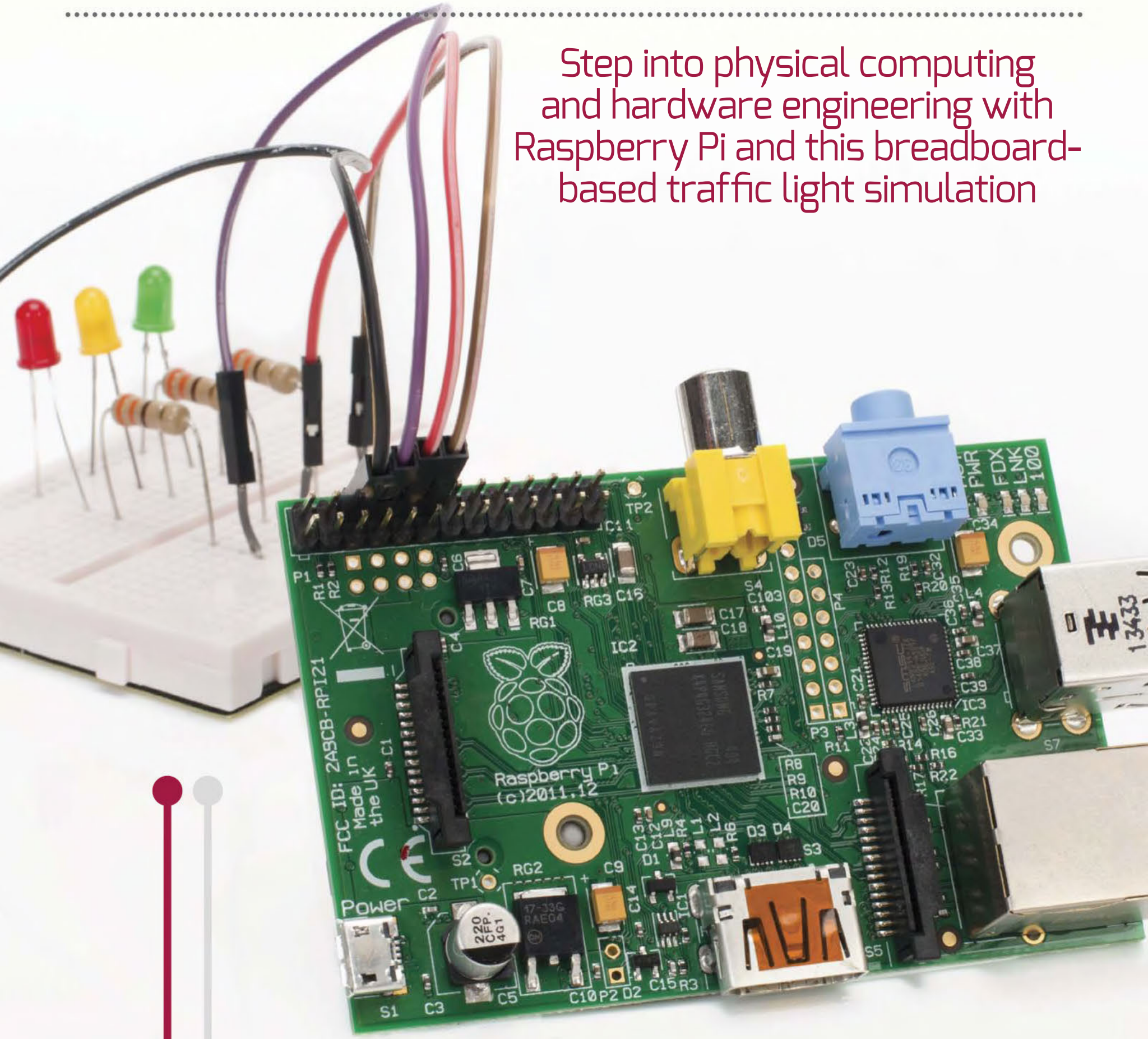
Yes, currently we only have the standard gun plugin, but if someone is familiar with electronics (and I think a lot of Raspberry Pi fans out there are) they can easily create their own plugins. I put a lot of effort into making the API user friendly. You can create, for example, an LED torch to replace one of the two gun attachments and you can then drive around in the dark wherever you want – under your bed, behind your wardrobe – and hide to create sniper's nests. Or you could create a simplified robotic arm for the back of the tank to throw other PiPanther tanks, knock them out of the game area. You can also make game elements like a landmine: you can create a small box that uses the same mechanism as the guns (infrared) and then put it in the arena so it explodes when people go near it.

"You could create a simplified robotic arm for the back of the tank to throw other PiPanther tanks, knock them out of the game area"



Simulate traffic lights using your breadboard

Step into physical computing and hardware engineering with Raspberry Pi and this breadboard-based traffic light simulation



Even with a platform as cheap, simple and powerful as the Raspberry Pi, taking your first foray into the world of physical computing and hardware projects can often be a fairly daunting experience. After unboxing your first Raspberry Pi, you might be a little intimidated by those GPIO ports! On the plus side, however, physical computing is perhaps the most rewarding way to experiment with computers and coding, as it allows you to manipulate tangible items in the physical world with the power of code.

One of the best ways to start making use of those GPIO pins and get really creative with your Pi later is a simple LED project, as you'll get to see the main GPIO commands in action. Let's make some mini traffic lights.

THE PROJECT ESSENTIALS

Breadboard

Male-to-female jumper cables

LEDs

330 Ohm resistors

01 Get the parts

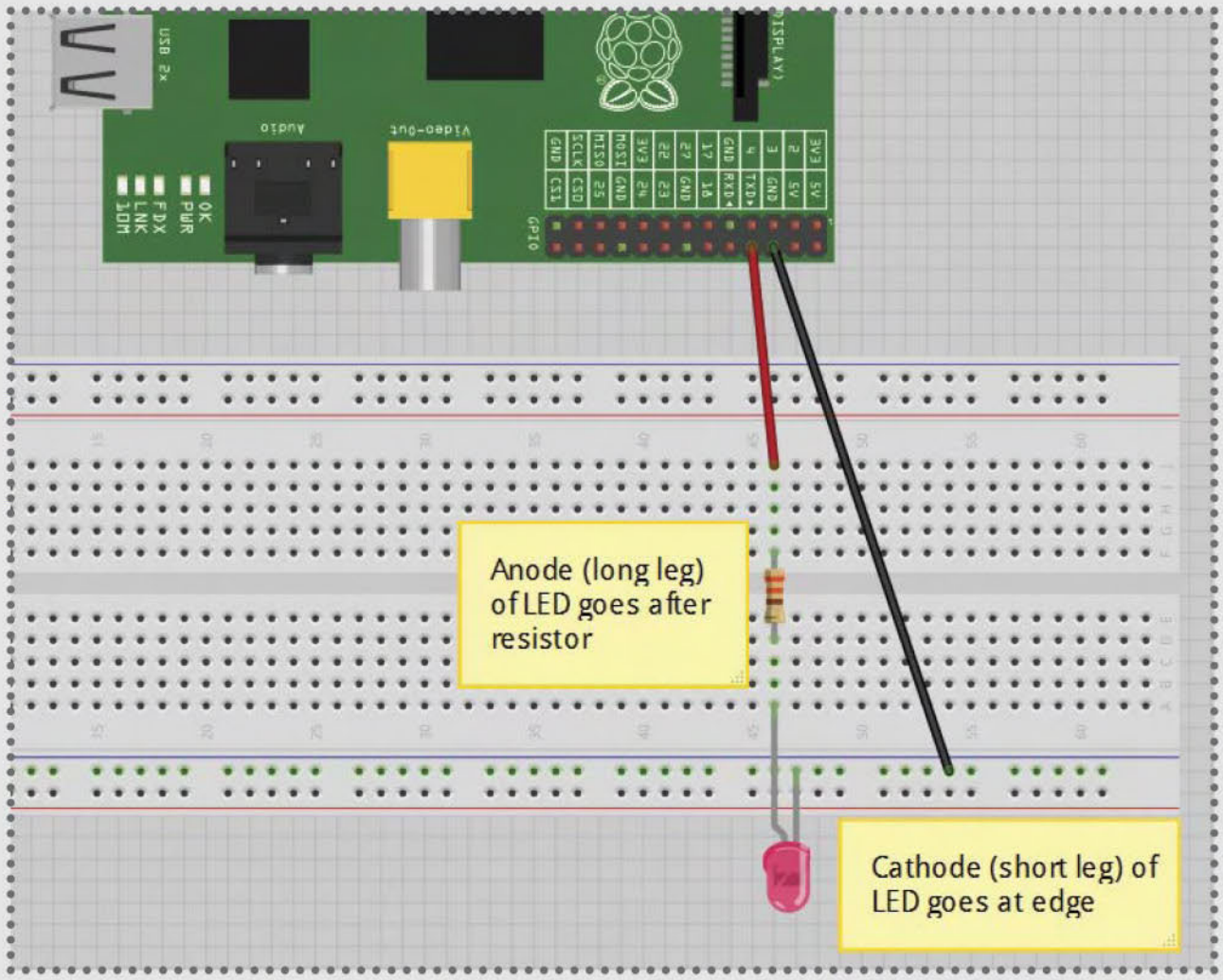
The parts for required for this tutorial should all be available from any reputable electronics dealer such as Maplin, CPC or one of the many independent Raspberry Pi retailers. The breadboard, LEDs, cables and resistors should cost you around £5 to £10 maximum.

> "The breadboard, LEDs, cables and resistors should cost you around £5 to £10 maximum"

02 Update your Pi

Before undertaking any project, regardless of whether it is hardware or software based, you should always update your Raspberry Pi to ensure you have the most stable software versions and latest functionality. This is especially important when experimenting with GPIO. You can do this using the following commands:

```
sudo apt-get update
sudo apt-get upgrade
```



Left You're after the pins on the outside edge of the Raspberry Pi, third and fourth from the corner

03 Prepare the hardware

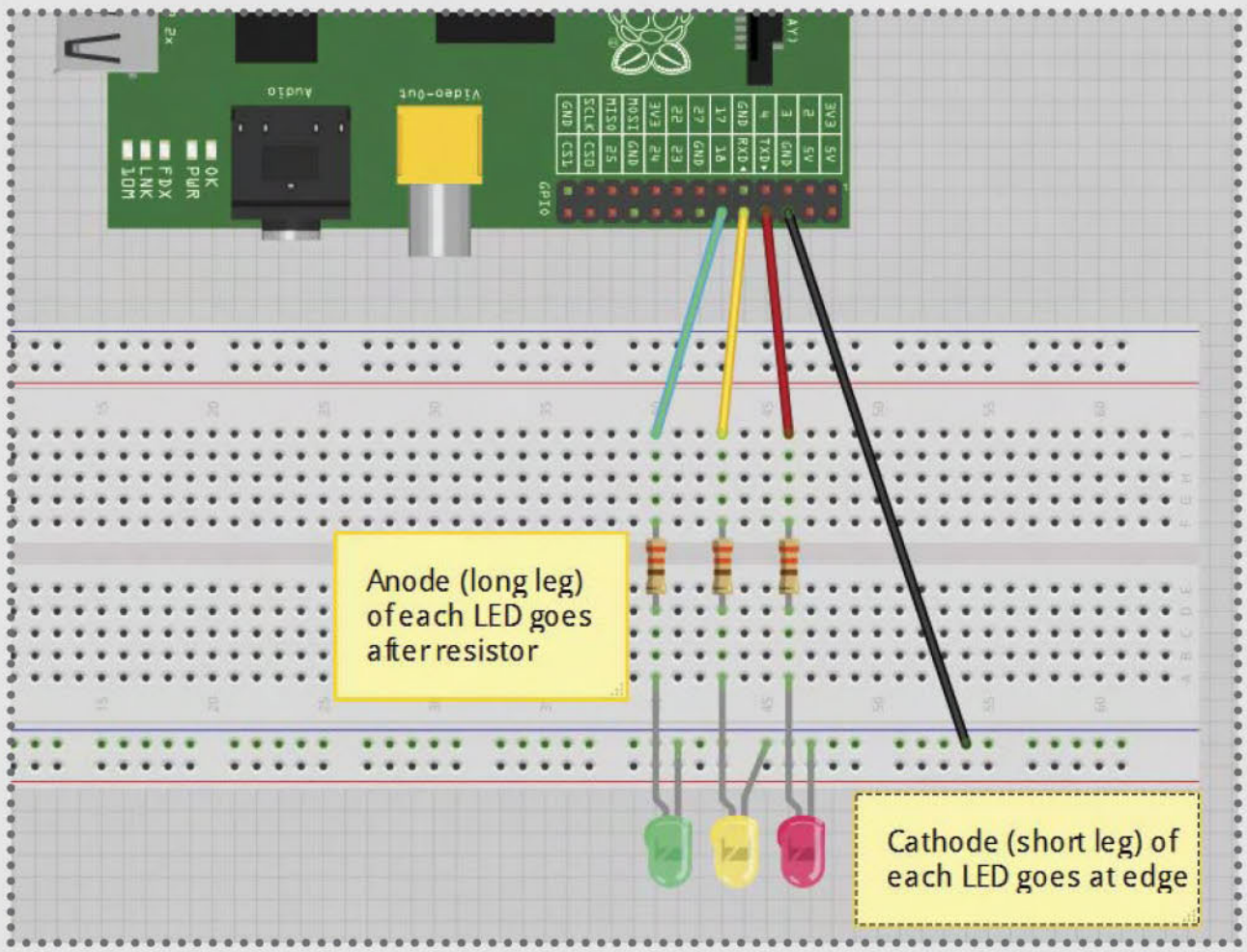
To begin, build and test a simple circuit to ensure everything is working correctly. Grab the breadboard, two jumper wires, the red LED, one 330 Ohm resistor and your Raspberry Pi, and connect everything together as shown in the above image. Ensure your Raspberry Pi is powered off when connecting things to the GPIO to avoid damage.

04 Add some code

The code shown on the following page is a simple snippet to test that the connection is correct and that the LED is working. Create a blank Python script by typing `sudo nano test.py` at the command line, then write in the code shown on the next page. Save and exit the script by using Ctrl-X and then pressing Y and Enter. You can then execute the script by typing `sudo python test.py` at the command line.

"Ensure your Raspberry Pi is powered off when connecting things to the GPIO to avoid damage"

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(14, GPIO.OUT)

GPIO.output(14, True)
time.sleep(10)
GPIO.output(14, False)

GPIO.cleanup()
```



Left To get the traffic lights up, simply extend the test circuit that we just made

05 Connect more LEDs

For a simulation of an actual traffic lights rig, we obviously need to add a green and yellow LED into the mix. Grab the LEDs and two more resistors, then connect everything together like in the image above, again ensuring the Pi is powered off first.

06 Add more code

In order to simulate a real block of traffic lights, we need to run through a sequence of colours. The code below achieves this, and the version we've uploaded to **http://bit.ly/1W6GdJp** is commented throughout so you can see what it is doing. As in Step 4, you will need to add this to a Python script to execute it, calling it 'trafficlight.py' this time instead of 'test.py' as before.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

Leds = [14,15,18]

for Led in Leds:
    GPIO.setup(Led, GPIO.OUT)
    GPIO.output(Led, False)

try:
  while True :

  GPIO.output(Leds[0], True)
  time.sleep(10)

  GPIO.output(Leds[1], True)
  time.sleep(2)

  GPIO.output(Leds[0], False)
  GPIO.output(Leds[1], False)
  GPIO.output(Leds[2], True)
  time.sleep(10)
```

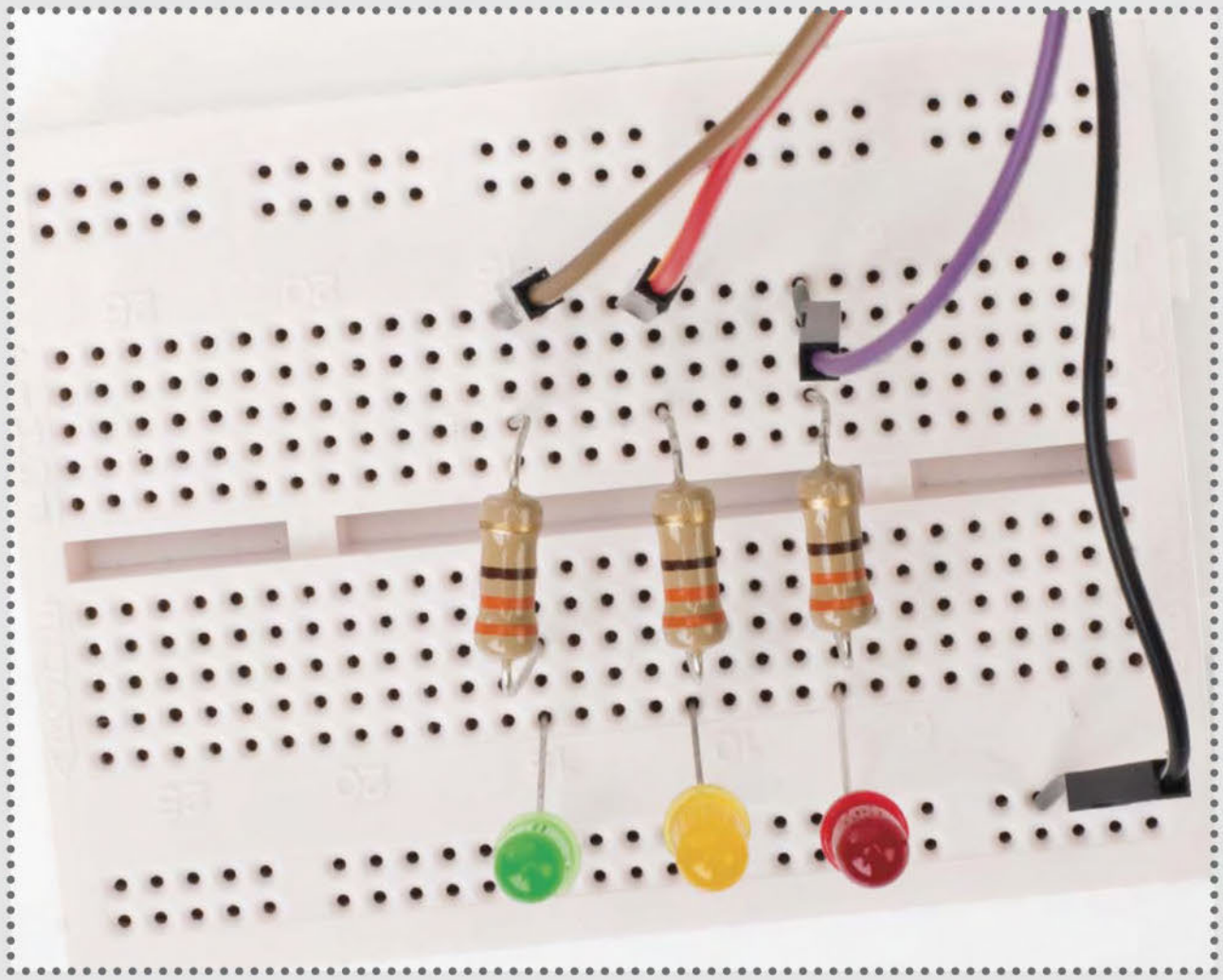
"To simulate a real block of traffic lights, we need to run through a sequence of colours. The code below achieves this"

```
    GPIO.output(Leds[1], True)
    GPIO.output(Leds[2], False)
    time.sleep(2)

    GPIO.output(Leds[1], False)

except KeyboardInterrupt:
    GPIO.cleanup()
```

07 Where to go next?

This was just a very simple introduction to experimenting with the GPIO on the Raspberry Pi. It shows how easy it is to manipulate things in the physical world with code and this is often the best way to learn. You can calways check out the Raspberry Pi Foundation resources section for more great ideas – just head to **www.raspberrypi.org/resources**.

"It shows how easy it is to manipulate things in the physical world with code and this is often the best way to learn"



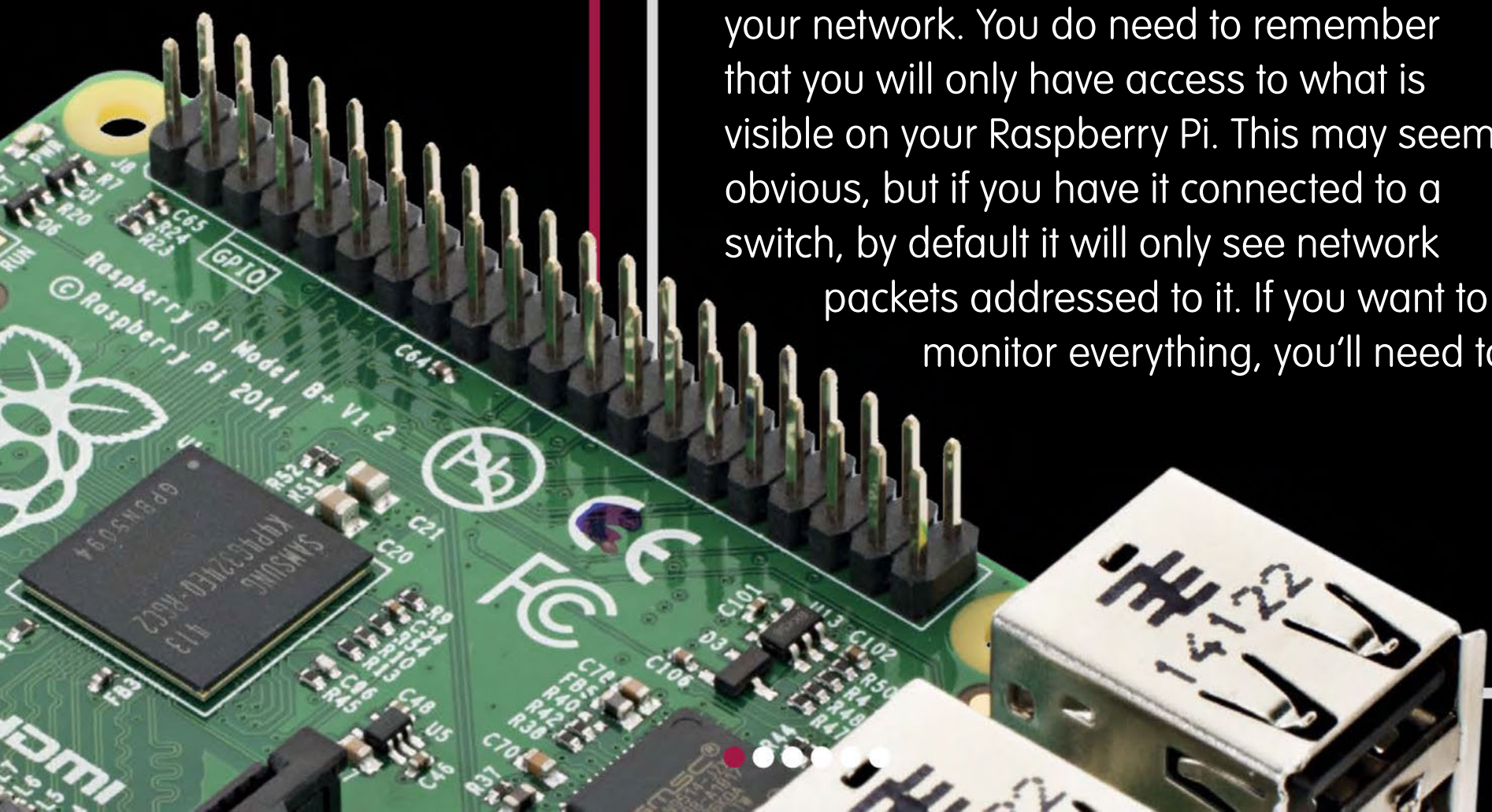
Left Now you can flash coloured LEDs in sequence, what about adding some user input and making a simple game?

Monitoring the activity on your home network

See what's happening on your network activity with Python and your Raspberry Pi

"We will use Python to look at what's available to code up your own tools and see what's happening on your network"

Raspberry Pis are great devices to use in monitoring applications. With the full set of IO pins, you can attach a range of devices to it, which leads to different environmental monitoring solutions. But what if you need to monitor something more technological, like network activity? This month, we will use Python to look at what's available to code up your own tools and see what's happening on your network. You do need to remember that you will only have access to what is visible on your Raspberry Pi. This may seem obvious, but if you have it connected to a switch, by default it will only see network packets addressed to it. If you want to monitor everything, you'll need to



configure your switch in order to mirror everything to the relevant port.

We'll start by looking at the lowest level available to Python. The standard Python library includes a module called 'socket'. By importing this module, you can talk to a network interface and observe what's happening on it. Remember that you will need root access to talk to physical devices, like the network interface. This means that you need to either be the root user or to run your Python code under sudo. First, you need to create a new socket object with the function: 'socket.socket(socket.AF_INET', 'socket.SOCK_RAW', 'socket.IPPROTO_TCP)'. The first parameter is the address family, the second is the socket type and the last is the protocol type. There is a full description of the options in the Python documentation. You can then bind this socket object to a specific address that represents the interface you're interested in. If you want to capture certain information, you can use the setsockopt() function of the socket object. This will enable you to capture the network packets addressed to you. However, if you want to capture everything visible, you need to set the interface to promiscuous mode. While you can do this within Python with the fcntl module, it can be messy. It's simpler to use:

```
sudo ifconfig eth0 promisc
```

If you're running your code as root, do this from within your program with:

```
import os
os.system("ifconfig eth0 promisc")
```

"Remember that you will need root access to talk to physical devices, like the network interface. This means that you need to either be the root user or to run your Python code under sudo"

You can now start collecting data, using the functions recv() and recvfrom(). They both take a number defining the buffer size. The difference is that recvfrom also includes the address that the data came from.

You can also use these raw socket objects to actively go out on the network. Assuming that ICMP packets aren't being blocked on your network, you can create a socket object with a prototype of IPPROTO_ICMP. Then you can use this to try and ping hosts on your network. The function connect() takes a tuple of a host and a port. If you can hit this machine at this port, the connect will succeed. Otherwise, it will fail with an error. Then send out an ICMP packet to this host. You will need to use the pack() of the Python struct module to create a binary struct that you can send on the socket connection – don't forget to close this socket before moving on. If you have specific services that you want to check, you can create a stream socket. Then connect this new socket to the host and port where the service is located and you can use the sendall() function to send some query. Use the recv() function to check the service's response and check services like email or web servers directly.

You may instead be interested in looking at the network activity happening on your Raspberry Pi. If so, look at the system information. If you don't want to work directly with socket objects, there are several modules that wrap this and give you higher level functions to look at various system metrics. One example is the psutil module. You can get a list of all the network connections on your Raspberry Pi with the function psutil.net_connections(). You can then count how many there are, or iterate over the list and query the elements for details like the remote address, the prototype or the status of the network

connection. If you are more interested in throughput numbers, you can get this data through psutil too. For this, the function in question is net_io_counters(). With no parameter, you will get the aggregate over all of the network interfaces. If you want it divided out, then you can pass in the parameter pernic=True. The returned object has the values of bytes_sent, bytes_recv, packets_sent and packets_recv. If you asked for the results broken down by network interface, you can get the list of interfaces with the function keys(). You can then pull the results for each interface using the keys.

Now you will hopefully have enough ideas and tools to add some basic network monitoring options to your own code. We have only been able to cover the basics here, so don't be afraid to do some research and experimentation to see what else you can do.

"If you are more interested in throughput numbers, you can get this data through psutil too. For this, the function in question is net_io_counters()"

Nagios

This month, we have looked at some examples of how to include network monitoring into your own program code. But, what if your primary intention is to do some serious monitoring and you still want to be able to use Python? A good choice might be nagios. Assuming that you are using Raspbian, you can install nagios with the command:

```
sudo apt-get install nagios3
```

This installs the monitoring tools, along with the web interface that you can use to control it. While nagios includes a large number of monitoring tools, you can use Python to write your own plugins and add more functionality. In order to do so, you will also need to install the NRPE server package with:

```
sudo apt-get install nagios-nrpe-server
```

You should put your Python scripts in the same directory (/usr/lib/nagios/plugins) as the system plugins. This simply makes configuration much easier. In order to trigger alerts within nagios, you will need to set an exit code with sys.exit(), where the possible values are:

Exit Code	Status
0	OK
1	WARNING
2	CRITICAL
3	UNKNOWN

Once you have your script written, you need to define it as a command in the file '/etc/nagios/nrpe.cfg', where you give it a command name and point it to the location on the file system. To add this new command to the checks, you will need to add a 'define' section to the file '/stc/nagios/objects/commands.cfg'.

"Once you have your script written, you need to define it as a command in the file '/etc/nagios/nrpe.cfg', where you give it a command name and point it to the location on the file system"

The Code


```
# The first step is to import the socket module
import socket

# You need a socket object to work with
# The first parameter is the address family
# The second parameter set the socket type to raw
# The third parameter sets the protocol to TCP
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
```

The Code NETWORK MONITORING

```
# The interface needs to be in promiscuous mode.
# If you forgot, you can use the following code
import os
os.system("ifconfig eth0 promisc")

# In order to see the outside world, you need
# to bind this socket to a physical interface
s.bind(('192.168.0.11', 0))

# You can start collecting data recvfrom include the source address
packet_data = s.recvfrom(65565)

# To check a web server, you can send a request to get the index page
s2 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s2.connect(('www.google.com', 80))
s2.sendall('GET index.html')

# You can get the response data with
data = s2.recv(10)

# With psutil, you can easily get both individual and aggregate network data
import psutil

# The network connections are given by
net_conns = psutil.net_connections()

# The number is
num_conns = len(net_conns)

# Aggregate data is give by
agg_stats = psutil.net_io_counters()

# The individual elements are
print "Bytes sent = ", agg_stats.bytes_sent
print "Packets received = ", agg_stats.packets_rev
```

Raspberry Pi Camera Robot

Dawn Robotics' cheaper and slightly less intimidating robot is a customisable Raspberry Pi-powered kit

Back in the robots issue, we reviewed the Rover 5 Seeeduino, a big old robot kit that had a lot of potential but a couple of problems with construction. This month we're looking at the Raspberry Pi Camera Robot from the same distributors (Dawn Robotics), a slightly cheaper, more reasonable robot kit that may be very familiar to those who saw our original Raspberry Pi robot feature way back in issues 2 and 3 (available in this app, if you missed them). It uses a similar chassis but has some massive improvements over the control method of that original robot, thanks to a mini Arduino board that's been added to the setup. It's a bit less DIY but it helps you get a better end result faster.

The Raspberry Pi camera bot comes in a couple of configurations, depending mostly on your budget. The basic kit comes with the chassis – which includes the motors and wheels – the Arduino control board and the camera pan-and-tilt mount to go on the front of the kit. There's also a lovely power regulator and a battery pack included, but you can also grab a pre-loaded SD card,

TECH SPECS

Locomotion
x2 DC motors

Controllers
Dagu Arduino Mini Controller and Raspberry Pi

Sensors
Raspberry Pi camera, speed sensor

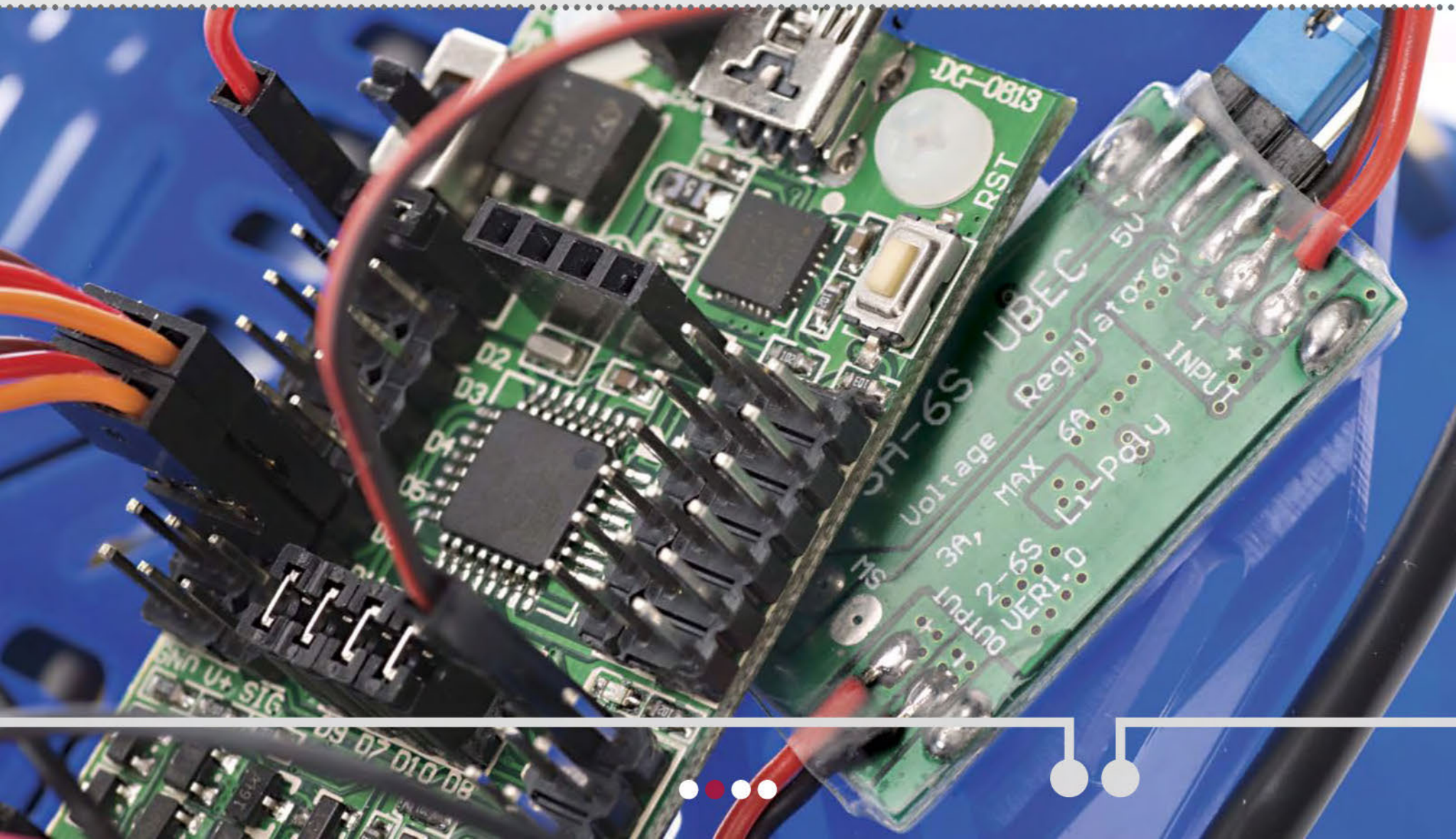
Chassis
Dagu Micro Magician V2

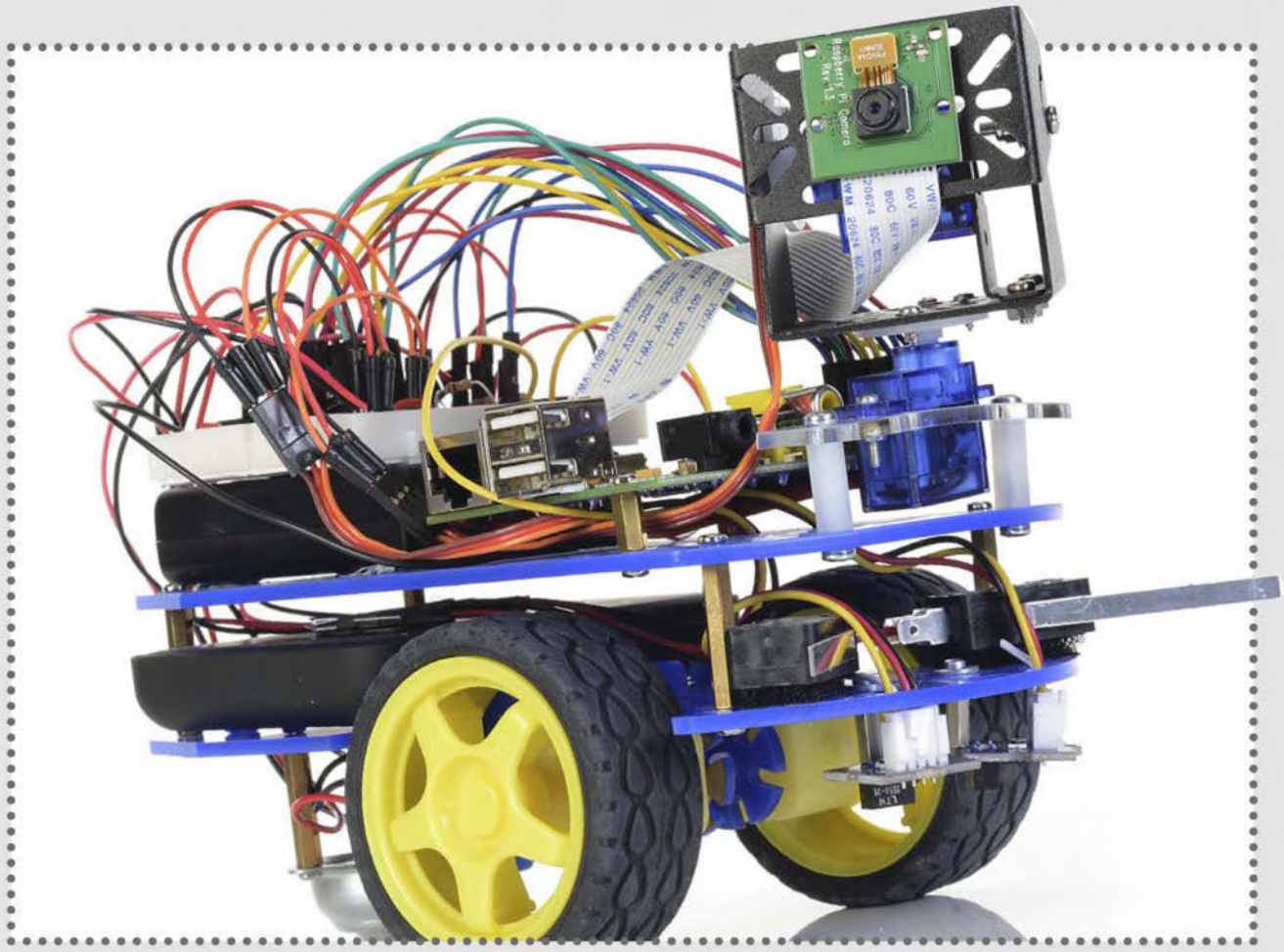
Dimensions
125 x 175 x180 mm

Weight
0.55 kg

Autonomy
Not out of the box, but remote control via browser

Price
£48.99 ($74)





Left The roomy Micro Magician V2 chassis has plenty of spots for additional controller boards

a recommended Wi-Fi adapter, power brick, Raspberry Pi and everything you would need to completely build the robot without needing any of your own kit. It's not necessary at all and it does increase the price, but it helps for those of varying skill level.

Building the actual robot is not always easy, due to the switching out of parts here and there (for a good reason), and not all the instructions are correct in the box, so you will need to hunt them down on the Dawn Robotics website. It can be very fiddly as well and takes a lot longer than you'd think to build: we spent a couple of hours on it. Generally, though, it's straightforward.

The final built machine is nice, simple and sturdy. There is a limited number of wires twisting around the chassis, yet there's plenty of room for adding to and upgrading the robot with extra sensors and functions and such. You can also use it quickly out of the box and connect to it as a wireless access point before immediately looking through the camera via a HTML interface. From here you can even control the robot via

Pros

Cheap and very little previous knowledge required – can basically work the minute you take it out of the box

Cons

Instructions could be slightly better and some parts are unnecessarily fiddly. It takes a little longer to build than you'd think

some virtual joysticks – one for movement and the other for camera control. This is optimised for smartphones with multi-touch capabilities but works fine with a mouse. The stream has very little lag, especially at close range, however in a building with busy Wi-Fi signals you'll be a little limited in how far away you can control it.

The programming is all Python and Arduino-based, available via the Dawn Robotics website, so you can easily program some autonomy into the robot and add it to the SD card plugged into the robot. It's all simple enough for more novice users, with plenty of advanced options for more veteran programmers and robotic enthusiasts.

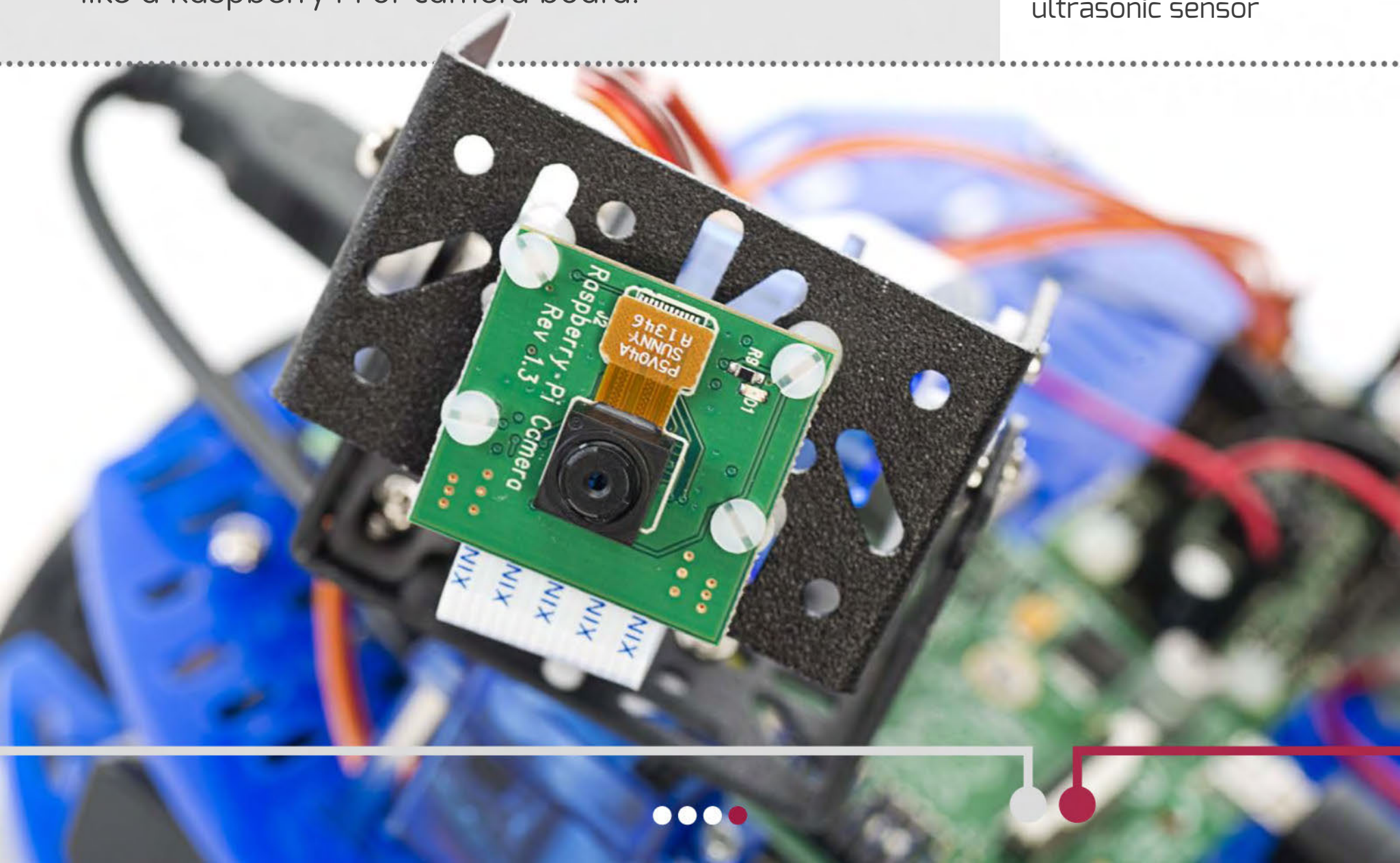
With no soldering involved and plenty of upgrading and programming options, this is an excellent kit for people getting into hobby robotics. The ease of construction and the way it works out of the box can be very helpful to users who might be intimidated by a robot project. It's also quite sturdy and allows you to use pre-existing components that you may have lying around, like a Raspberry Pi or camera board.

THE VERDICT

An excellent starter kit for a beginner, but one that can really be modified and upgraded to be a beast of a robot by others. The price point is right as well, and although some of the construction can be fiddly, it's a sturdy piece of kit that's great for many types of projects

More info:
www.dawnrobotics.co.uk

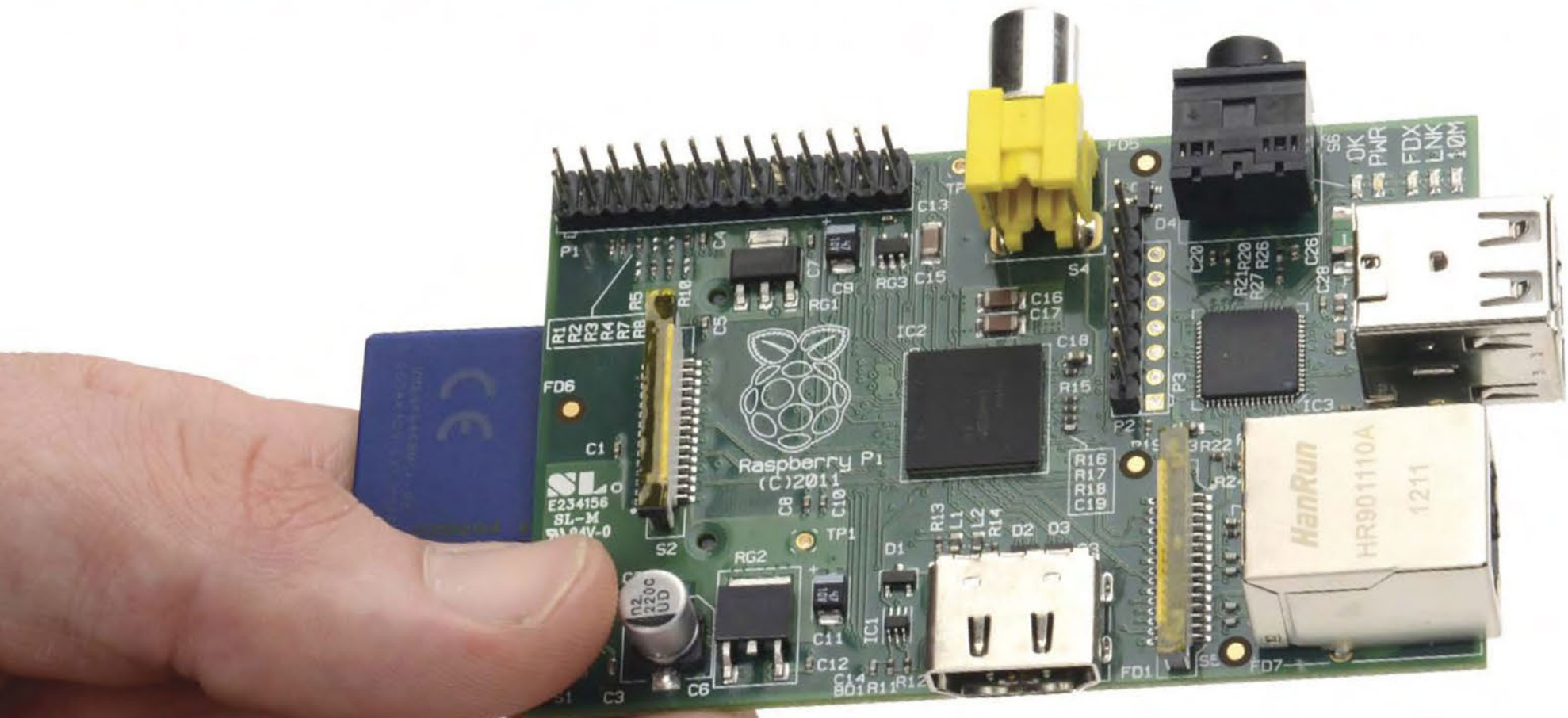
Below You can mount your Pi NoIR onto the head instead – or an ultrasonic sensor



Talking Pi

Join the conversation at...

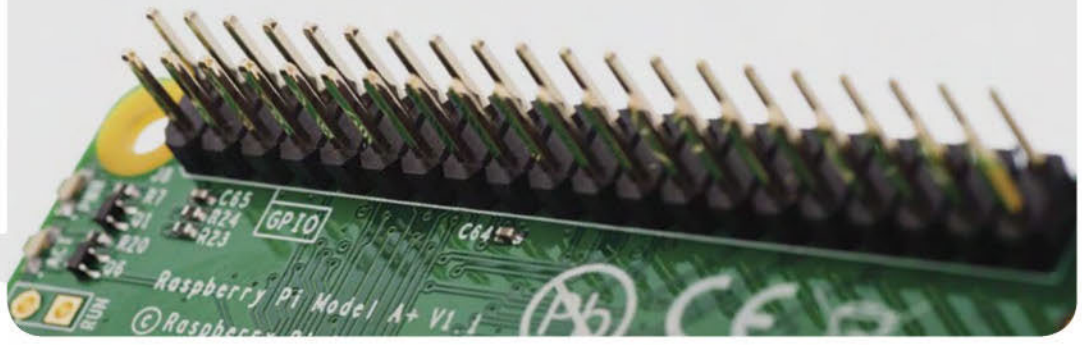
@linuxusermag | Linux User & Developer | @ RasPi@imagine-publishing.co.uk



The Raspberry Pi encourages a lot of hands-on work and this means it doesn't quite work like – or as easy as – your laser focus-tested smartphone interface, or even a normal computer. So we're answering your burning Raspberry Pi questions each issue – get in touch with us on Twitter, Facebook or by email.

The GPIO ports confuse me – are they all for general use or do they each have specific uses?
Will via email

If you're using them in the general purpose manner, most of them can be used in very basic ways, where as a handful also provide power (5V or 3V3) and a ground for that power. Those ones can't be changed. Some, however, also have specific functions that can be used instead of the general purpose one, such as a clock function or I2C inputs. These are required for the PiTFT, for example.



Keep up with the latest Raspberry Pi news by following @LinuxUserMag on Twitter. Search for the hashtag #RasPiMag

JUST A SCORE
WHAT'S YOUR JUST A SCORE?

Have you heard of Just A Score? It's a new, completely free app that gives you all the latest review scores. You can score anything in the world, like and share scores, follow scorers for your favourite topics and much more. And it's really good fun!

I'm building an arcade cabinet at the moment. What's the best way to run the games?
Tom via Facebook

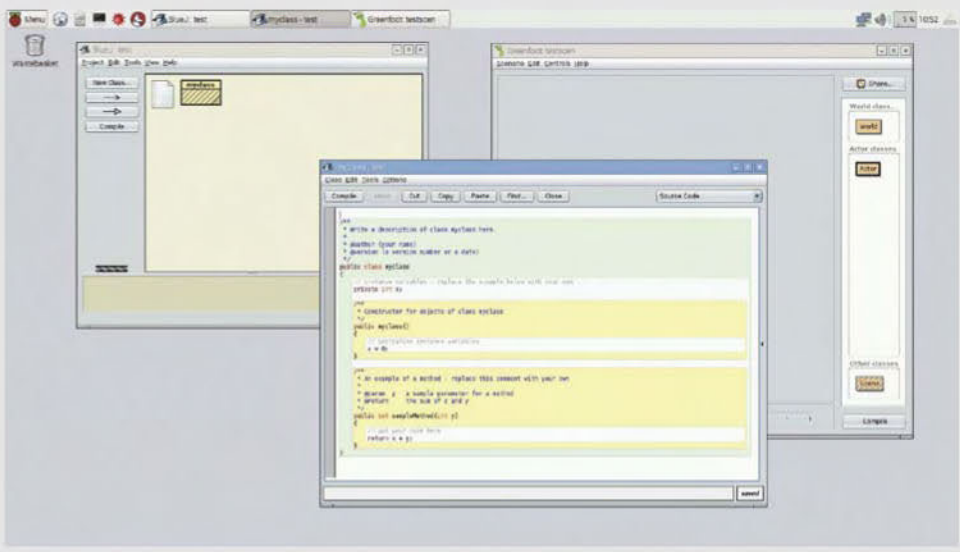
Try using the emulation distro called RetroPie; it has both the MAME and DOSBox emulators already configured, along with a load of emulators for old consoles such as the NES, SNES, Megadrive, etc. This distro also works a little better with any USB controllers you might have, including those for the PS3 or XBox 360 consoles. The benefit here is that it's also small and requires less power, making your arcade machine much more efficient.

I want to take my sister to a Raspberry Jam because they look really fun. Are there any in the West Midlands?
Jenny via email

There's one coming up in Birmingham, actually – it's on Saturday 5th December at the Google Digital Garage, which is just on Level 1 of the main library on Broad Street. Here's the event page: **bit.ly/1OS66fx**. The best place to look for Raspberry Jams is the Foundation's event tracker – there's a big map showing you where all the upcoming Jams are, so you can just look for the ones nearby: **raspberrypi.org/jam**.

Do I need to use a fresh SD card to upgrade to Raspbian Jessie?
Ross via email

We definitely recommend it, and so does the Foundation – head to **raspberrypi.org/downloads/raspbian** to find the Raspbian Jessie zip file along with the installation instructions. If you really don't want to use a fresh SD card, there is a workaround you can try but it might cause problems: **bit.ly/1NTYbji**.



Next issue

Get inspired **Expert advice** **Easy-to-follow guides**

BUILD A CAR COMPUTER

Get this issue's source code at:

www.linuxuser.co.uk/raspicode